International Center for Numerical Methods in Engineering
Master of Sciences in Computational Mechanics
Erasmus Mundus Master Course

Universitat Politècnica de Catalunya, Barcelona
Swansea University, Swansea

A two-phase Navier-Stokes flow with surface tension modeled in X-FEM

by

Hannes Schümann

Thesis Proposal

Advisor: Esther Sala Lardies
Pedro Díez Mejía

Barcelona, 28/06/2010

Abstract

A two-phase Navier-Stokes flow with surface tension modeled in X-FEM

Hannes Schümann

A method to calculate multiphase flows of two immiscible fluids with a zero-thickness interface approach is presented, which is also can handle the effects of surface tension.

For this, the incompressible Navier-Stokes-equations are solved in a two dimensional or axis symmetric domain of polygonal shape. An extended finite element discretization is applied to handle the discontinuities of the velocity and pressure solution without mesh refinement. The interface position is determined by interface capturing method with level sets, including the techniques of level set maintenance. The interface curvature for the surface tension force is determined by a piecewise polynomial approximation.

For the validation of our code, we present various examples: A steady example with analytical solution, a pure Stokes flow and the Reynolds-Taylor-instabilities are used to demonstrate the aspects of the code. Furthermore, for rising bubbles example we show a general agreement to results of numerical and physical experiments and also emphasize the improvement of the solution by enrichment. Additionally, the merging and separation of interface is demonstrated for an examples of two bubbles rising to a free surface. Finally, to give an outlook on possible industrial applications, the results for a drop separation in an inkjet printing process is presented.

Acknowledgments

I want to express my thanks to my supervisors of this thesis, Esther Sala Lardies and Pedro Díez Mejía. For finding the way through the variety of possibilities, I could trust on their support and guidance.

Also I want to thank my fellow students from the Masters Course, especially my mates Sébastien, Shi Yee and Sujith. By your support, discussions and questions I learned a lot.

Finally, grateful thanks to my parents, which supported me in any possible way and stand by my side. Danke.

Contents

List of Figures

Chapter 1

Introduction

1.1 Multi-Phase flow with surface tension

In everyday life we observe fluid phenomena with multiple phases and the effects of surface tension, like in a small droplet on a leaf. But also in industrial processes multi-phase flows appear, which may be highly characterized by surface tension. One could think of inkjet printing devices or fuel injection in combustion devices as examples of microscopic two -phase problems. Also, in combination with a heat-transfer module, the industrial metallic molding processes are two-phase flows determined by surface tension. Finally for high-performance yachts for sports, the waves on the free water surface are taken into consideration in the loads acting on the hull. For further development, proper simulation tools help to understand the phenomena taking place in these problems.

For many years, research on numerical methods concentrated on one-phase flow phenomena, but as numerical capacity of computers and super-computers rises continuously, more advanced topics of multi-fluids came into the focus. So the question of simulating multi-phase flows is a popular research topic and many papers have been

published on this. The main challenge of the two-phase flows is the handling of the discontinuities in the solution. On the one hand, the velocity becomes non-smooth due to the jump the in viscosities. And on the other hand, by surface tension, the pressure gets a jump-discontinuity at the interface.

The following section gives a short overview on some of the techniques for simulating multi-fluid flows.

1.2 Numerical models

The structure of this section mainly follows the good overview on numerical methods for interfacial flows given in (Smolianski, 2001). For this, we want to discuss the advantages and disadvantages of the most popular methods for simulating multiphase-flows.

1.2.1 Lagrangian methods

In Lagrangian methods, each of the discretization elements represents a specific part of the fluid and while the fluid flows, the mesh has to follow this movement. By this, the distinguishing of the two phases of a fluid is very accurate, since to each of the elements, exactly one set of fluid properties can be applied. Also, the non-linear convective term in the fluid momentum equations does not exist. But on the other hand, as the mesh follows the fluid, the mesh becomes irregular soon and gets highly distorted. This would lead to high numerical inaccuracies. Also, a high change of the topology of the interface, like separation, needs complicated mesh distortion methods.

1.2.2 Free Lagrangian methods

In free Lagrangian methods, the mesh is reinitialized and/or reshaped regularly. By this, the mesh quality is improved compared to pure Lagrangian methods. But still, topological changes of the interphase would introduce high numerical effort and could not be imposed directly.

1.2.3 Meshless particle-based methods

To circumvent the problems introduced by remeshing and mesh distortion, in particle-based methods the mesh is completely abandoned. In these methods, to each of the particles, a certain amount of fluid is applied to. By this, each *particle has a set of attributes, like mass, position, velocity, momentum and energy* (Smolianski, 2001). The interaction of the particles defines the state of the fluid; so for this, reliable interface identification algorithms are needed. This interface identification sets the quality of the representation of the interface.

1.2.4 Mixed Eulerian-Lagrangian methods / ALE

In these methods, the Lagrangian movement of the mesh is combined with the Eulerian formulation of the fluid moving in a fixed mesh. The nodes of the mesh are adapted to the movement of the fluid, which could mean completely following the fluids movement, but the mesh could also be just slightly moved. The nodes move arbitrary, as already indicated by the full name: Arbitrary Lagrangian-Euler. If the change of the topology of the interface is within certain restrictions, this method can represent two-phase flows with high accuracy. This is why it is applied in many multiphase problems to free-surface flows. But for large changes of the interface topology, again, the mesh adaptation would need high programming and computational effort.

1.2.5 Eulerian methods

In Eulerian methods, the fluid moves through a fixed mesh. By this, the mesh is independent from the topology of the interface and this method is commonly used for general flow problems. For this formulation, a high variety of methods have been developed for multiphase flows.

Surface tracking

For surface tracking methods, the position of the interface itself is saved and transported over time. They may be saved in terms of a distance above a certain line, which limits the topological shape of the interface dramatically. Or the interphase position may be saved in form of a parametrical representation. For latter, high changes in the interphase topology would need complicated geometrical treatment of the interphase points. One advantage is, that the position of the interface may be saved on a smaller length scale than the computational mesh. This enables a very fine resolution of the interphase geometry, such as curvature and normal direction. But nevertheless, the programming effort for large topological changes may become very high. Also, if the information of the interface accumulates in certain areas, the interface representation may lack accuracy elsewhere.

Volume tracking

Volume tracking methods work with markers for the cells and the interphase itself is reconstructed when necessary. The markers are also transported over time and by this, the high topological changes of the interface shape can easily be handled. But since the actual interphase position can only be derived from finding marked cells with neighbors being not (or differently) marked, the actual interface geometry representation is relatively poor. Also, it may happen that the markers accumulate in certain areas in the domain, while other areas lack information about the interface. Also, the transport of the markers may be a relatively costly process.

Interface-capturing

For the interphase-capturing, an additional scalar field defines the position of the interface in the domain. This scalar field may be discontinuous, how it is used in the volume-of-fluid approach (VOF), but also a continuous function, how it is done with level set methods.

In the VOF-methods, the phases are defined to be "one" for one phase and "zero" for the other. This discontinuous function is transported with the fluid velocity as advection velocity. By this, the change of position of the interphase can be tracked. This method became very popular, due to the ability of natural handling of topological changes of the interface and its good mass conservation properties. But the disadvantage of this method is that a discontinuous function has to be advected. Secondly, the interface itself is only defined by the discontinuity-line of the VOF-function, which makes it difficult to find the exact position and by this, the exact curvature and normal direction of the interface.

In contrast to this, the level set approach works with a continuous function, which is initialized as a signed distance function, being positive in one domain and negative in the other. As in other methods, this level set function is transported according to the fluid velocity. But here, since a continuous function is transported, the convection problem is easier to solve. Also, as long as the level set function is close to be a signed distance function, the curvature and the normal direction can be found naturally. The disadvantage of the continuous formulation turned out to be that it is less accurate than VOF methods in terms of mass conservation and that there is numerical smearing of the interphase geometry. Also, since it is necessary to keep the level set function close to a signed-distance function, reinitialization becomes necessary, what may additionally corrupt the mass-conservation properties.

The methods mentioned in the section before had been the most important and most commonly used methods on multiphase flows. Of course, it just can be an overview on the existing methods, and can not be a complete list of all methods ever used for this kind of problems.

1.3 Numerical strategy and scope of the thesis

Taking the advantages and disadvantages of the methods in the previous section into account, we decided to model our two-phase flow in a pure Eulerian manner, meaning the fluid will flow through a fixed mesh.

As mentioned, the discontinuous physical properties of the two fluids introduce discontinuities into the flow solution. With standard Finite Element Methods (FEM), those discontinuities would not be clearly resolved and a very high mesh density close to the interface is required to represent the high gradients properly. Instead, we choose an enriched version of FEM, which is naturally able to handle discontinuities, by using discontinuous shape functions where needed. The so called *eXtended Finite Element Method* (X-FEM) was firstly mentioned by (Moes et al., 1999) in solid mechanics, in the context of crack growth, but can also be applied to the discontinuities in multiphase flows. In this, specific discontinuous shape function discretize the velocity and pressure solution of those elements, which are intersected by the interface.

For interphase localization we choose the continuous interface capturing method with a level-set function. For calculating the curvature and the normal direction, we choose a polynomial approximation of the interface. By this, we hope to become more independent on the level sets' property of being a signed distance function and therefore use the reinitialization less often.

The base of the code presented here, is an XFEM code of Esther Sala-Lardies[1] which initially solved the two-dimensional steady Stokes problem for a fixed interface.

[1]Laboratori de Càlcul Numèric, Universitat Politècnica de Catalunya, Barcelona

The velocity and the pressure are solved fully coupled. To this, the following features were added:

- The additional terms for a Navier-Stokes-flow (mass and convection terms) were introduced.
- A forcing term on the right-hand side of the system was added, which represents the surface tension. The surface tension is dependent on the interface normal direction and the curvature, which we calculate with the help of a piecewise polynomial approximation.
- The fixed signed distance function in the initial code was changed to a level set function, which is convected with the fluid velocity.
- To have a – up to certain limit – stable code, a stabilization method of solving the Navier-Stokes equations and maintenance of the level set function (reinitialization as well as smoothing) were added.
- Additionally to the two-dimensional formulation of the initial code, we introduced the possibility of calculating axis-symmetric problems.

1.4 Outline of the thesis

This first chapter was devoted to give an overview on the type of problems, which this thesis deals with. The most commonly used methods for this had been discussed.

In Chapter 2, the physical principles acting in multiphase flows are presented and also, which model is used in this thesis for their simulation. This is done for both, the two-phase flows in general, but also specifically for the effect of the surface tension.

In Chapter 3 the way how we solve the problem is discussed in detail. This contains the spatial discretization and the system of equations (Section 3.6), the approximation

of the forces at the interface (Section 3.8) and the domain detection / interface localization by the level set function and its maintenance (Section 3.10 and Section 3.11)

To show the general properties of the code and also its accuracy, the code had been applied to several examples. The discussion of this can be found in Chapter 4. In there, a wide range of examples we applied this code to is presented: From simpler, steady and non-physical examples with analytical solution to a Stokes problem, the Rayleigh-Taylor instability and Rising bubbles problems. Also, as it was a key-point in the discussion before of the advantages and disadvantages of the possible methods, we present the codes' ability to deal with strong topological changes in Section 4.5.

Moreover, to give a first hint for the industrial application of the code, an example of the ejection of droplets in inkjet plotting process is given in Chapter 5.

Finally, a short summary of the results is given and future aspects on the accuracy and efficiency of the code are discussed in Chapter 6.

Chapter 2

Physical Model

2.1 Physical description

2.1.1 Flow model

For our flow model, we assume an unsteady laminar flow of two immiscible fluids. Both fluids are assumed to be viscous Newtonian fluids. Also, we do not consider temperature changes in the domain, thus there are no thermal viscosity and density changes. Moreover, we restrict our examples to very small Mach numbers ($Ma < 1/3$), which is the most important condition of a flow to be incompressible.

By this isothermal and incompressible restrictions, we may consider constant viscosity and density within each of the fluids. Nevertheless, the density and viscosity of the two fluids may differ from each other.

2.1.2 Interface conditions

Over the last two centuries, two main different models of the physical conditions at the interface have been developed. The primary assumes the interface itself to be of zero thickness ("Sharp interface"), in which the density and viscosity have a jump discontinuity. In later, more exact investigations it was found that the transition of the physical parameters happens rapidly, but smoothly on the interface. But still, the actual thickness of the interface is hard to determine. Also the zero-thickness approach describes the phenomena on the interface accurate enough, as long as the geometrical scale of the effects of interest is much bigger than the interface thickness.

We choose to use the zero-thickness assumption for our interface-model. By this, the viscosity and density have a jump discontinuity along the interface. Also, we assume that there is no mass transfer through the interface. Finally we take surface tension into consideration. For this, a constant surface tension coefficient is assumed.

2.2 Governing equations and surface tension

The flow we characterized is calculated for a bounded region of interest, which is our domain Ω. The boundary of our domain $\partial\Omega = \Sigma$ is described by a polygon. For each of the fluids, a subdomain Ω_1 and Ω_2 of arbitrary shape are defined and the interface between the fluids is Γ. So the domain Ω is union of the two subdomains and the interface $\Omega = \Omega_1 \cup \Omega_2 \cup \Gamma$. The two subdomains may have multiple connections and therefore the interface may be divided into multiple sections and the interface may intersect the domain boundary Σ. A principal scheme of this is shown in Figure (2.1)

The flow regime characterized above can be fully described by the momentum and mass conservation equations of the incompressible Navier-Stokes-Equations, with an

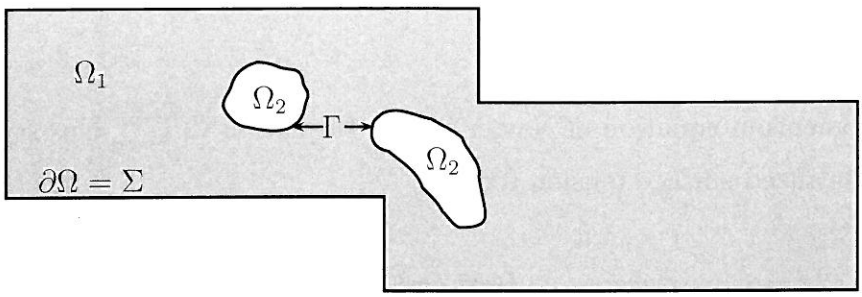


Figure 2.1: Schematic of domain, subdomains, boundary and interphase

additional force term representing the surface tension.

$$\rho_i \frac{\partial \boldsymbol{u}}{\partial t} + \rho_i \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \nabla \cdot \left(2\mu_i \underline{\underline{\mathbf{S}}}\right) + \nabla p = \rho_i \boldsymbol{g} + \mathbf{f}_\Gamma \quad \text{in } \Omega_i, i = 1,2 \tag{2.1a}$$

$$\nabla \cdot \boldsymbol{u} = 0 \quad \text{in } \Omega_i, i = 1,2 \tag{2.1b}$$

with the unknown fluid velocity $\boldsymbol{u}(\boldsymbol{x}, t)$ and the unknown pressure $p(\boldsymbol{x}, t)$. $\underline{\underline{\mathbf{S}}}$ is the deformation rate tensor, defined like $\underline{\underline{\mathbf{S}}} = \frac{1}{2}(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T)$. The physical fluid parameters density and viscosity are fixed for each fluid: ρ_i and μ_i of the i-th fluid ($i = 1, 2$). The gravitational body forces are introduced by the forcing term, with $\boldsymbol{g}$ being the constant gravity vector.

Also, sufficient boundary conditions have to be added.

2.2.1 Surface tension forcing term

For the forces along the interface, we choose a standard model, that assumes that the jump of the normal stress along the interface (denoted as $[\sigma \mathbf{n}]_\Gamma$) is proportional to the local curvature $\kappa = \kappa(\boldsymbol{x}), \boldsymbol{x} \in \Gamma$. Despite many others, this is also used in (Groß, 2008) and the full derivation of it can be read in (Smolianski, 2001).

$$[\sigma \mathbf{n}]_\Gamma = \tau \kappa \mathbf{n}_\Gamma$$

Here, τ is the constant surface tension coefficient, κ the local curvature $\kappa = \kappa(\boldsymbol{x})$ and $\mathbf{n}_\Gamma$ the outer unit normal of the interface.

In the momentum equation of Navier-Stokes (Equation (2.1a)), this stress balance becomes a localized surface tension force:

$$\mathbf{f}_\Gamma = \tau\kappa\delta_\Gamma \boldsymbol{n}_\Gamma$$

where the Dirac δ-distribution of δ_Γ concentrates the forcing term directly on the interface.

In (Smolianski, 2001) it is shown that the tangential stress is continuous across the interface and that for inviscid flows, the normal stress conditions balances exactly the pressure jump, which is indicated by the Young-Laplace equation

$$\Delta p = \tau\kappa$$

Chapter 3

Numerical Approach

Adding the boundary conditions, the complete strong formulation of the problem above can be read like: With given physical parameters density ρ_i, viscosity μ_i, surface tension coefficient τ and the gravity $\boldsymbol{g}$, and the Dirichlet conditions $\boldsymbol{u}_D$ on the domain boundary Σ, find the velocity field $\boldsymbol{u}$ and the pressure field p, such that:

$$\begin{aligned} \rho_i \tfrac{\partial \boldsymbol{u}}{\partial t} + \rho_i \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \mu_i \nabla \cdot \left(\nabla^S \boldsymbol{u}\right) + \nabla p &= \rho_i \boldsymbol{g} + \tau \delta_\Gamma \kappa \boldsymbol{n} && \text{in } \Omega \\ \nabla \cdot \boldsymbol{u} &= 0 && \text{in } \Omega \\ \boldsymbol{u} &= \boldsymbol{u}_D && \text{in } \Sigma = \partial\Omega \end{aligned} \tag{3.1}$$

Velocity boundary conditions are imposed everywhere on $\partial\Omega$. Additionally, as the flow is confined, one pressure degree of freedom has to be fixed to a discrete value.

As described in Chapter 2, the method we choose is the X-FEM, the extended finite elements method. A full and detailed description of the features of the code are explained in the following chapter.

3.1 Time discretization

For the time derivative in the first part of the problem mentioned before, we use a first order accurate Backward Euler discretization:

$$\frac{\partial \boldsymbol{u}}{\partial t} = \frac{\boldsymbol{u}^{n+1} - \boldsymbol{u}^{n}}{\Delta t}$$

where $\boldsymbol{u}^{n+1}$ is the unknown velocity of the current time step and $\boldsymbol{u}^{n}$ is the velocity of the time step before, which is known. So the full system is solved for $\boldsymbol{u}^{n+1}$ and p^{n+1} in a coupled way at the same time.

In standard fully implicit methods, problems with the convection term $\boldsymbol{u}^{n+1} \cdot \nabla \boldsymbol{u}^{n+1}$ would occur, since this introduces the solution of a non-linear system. To circumvent solving the non-linear system, we choose to approximate this term with the velocity of the time-step before, so the convection term becomes $\boldsymbol{u}^{n} \cdot \nabla \boldsymbol{u}^{n+1}$. The other terms on the left hand side are calculated for the current time-step $n+1$, and therefore the time-stepping scheme is semi-implicit. So, with the time discretization we choose, the momentum equation of the problem reads like:

$$\begin{aligned} \frac{1}{\Delta t}\rho_i \boldsymbol{u}^{n+1} + \rho_i \boldsymbol{u}^{n} \cdot \nabla \boldsymbol{u}^{n+1} - \mu_i \nabla \cdot \left(\nabla^S \boldsymbol{u}^{n+1}\right) + \nabla p^{n+1} \\ = \frac{1}{\Delta t}\rho_i \boldsymbol{u}^{n} + \rho_i \boldsymbol{g} + \tau \delta_\Gamma \kappa \boldsymbol{n} \quad \text{in } \Omega \end{aligned}$$

Also, the $n+1$-solution has to fulfill the incompressibility conditions and the possibly time-dependent Dirichlet boundary conditions:

$$\nabla \cdot \boldsymbol{u}^{n+1} = 0 \quad \text{in } \Omega$$

and

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}_D(t) \quad \text{in } \Sigma = \partial\Omega$$

3.2 Weak form

As known, finite element methods do not solve the system of partial equations directly, but in the so-called weak form. This chapter derives the formulation of our problem at hand in the weak form and it mainly follows (Zlotnik, 2008)

To get the weak formulation of the problem above, it is multiplied with the test function $\boldsymbol{w}$, which is defined like $\{\boldsymbol{w} \in \mathcal{H}^1(\Omega) | \boldsymbol{w} = 0 \text{ on } \Sigma\}$, and also integrated over the full domain Ω:

$$\begin{aligned}
&\int_\Omega \boldsymbol{w}^{n+1} \cdot \rho_i \boldsymbol{u}^{n+1} \, \mathrm{d}\Omega + \int_\Omega \boldsymbol{w}^{n+1} \cdot \rho_i \boldsymbol{u}^n \cdot \nabla \boldsymbol{u}^{n+1} \, \mathrm{d}\Omega \\
&- \int_\Omega \boldsymbol{w}^{n+1} \cdot \left(\mu_i \nabla \cdot \left(\nabla^S \boldsymbol{u}^{n+1}\right)\right) \, \mathrm{d}\Omega + \int_\Omega \boldsymbol{w}^{n+1} \cdot \nabla p^{n+1} \, \mathrm{d}\Omega = \\
&\frac{1}{\Delta t} \int_\Omega \boldsymbol{w}^{n+1} \rho_i \boldsymbol{u}^n + \int_\Omega \boldsymbol{w}^{n+1} \cdot \rho_i \boldsymbol{g} \, \mathrm{d}\Omega + \int_\Omega \boldsymbol{w}^{n+1} \cdot \tau \delta_\Gamma \kappa \boldsymbol{n} \, \mathrm{d}\Omega
\end{aligned}$$

Since the momentum equation is (mainly) calculated for the current time step, the test function $\boldsymbol{w}$ belongs also to the time step $n+1$. Additionally, by the definition of the Delta function δ_Γ, the integral of the surface tension force becomes a localized line integral:

$$\int_\Omega \boldsymbol{w}^{n+1} \tau \delta_\Gamma \kappa \boldsymbol{n} \, \mathrm{d}\Omega = \int_\Gamma \boldsymbol{w}^{n+1} \tau \kappa \boldsymbol{n} \, \mathrm{d}\Gamma$$

Integration by parts is applied to the viscosity term:

$$\begin{aligned}
&\int_\Omega \boldsymbol{w}^{n+1} \cdot \left(\mu_i \nabla \cdot \left(\nabla^S \boldsymbol{u}^{n+1}\right)\right) \, \mathrm{d}\Omega = \\
&\int_\Omega \nabla \cdot \left(\boldsymbol{w}^{n+1} \cdot \mu_i \nabla^S \boldsymbol{u}^{n+1}\right) \, \mathrm{d}\Omega - \int_\Omega \nabla \boldsymbol{w}^{n+1} : \mu_i \nabla^S \boldsymbol{u}^{n+1} \, \mathrm{d}\Omega
\end{aligned}$$

with the notation:

$$\nabla \boldsymbol{u} : \nabla \boldsymbol{v} = \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{du_i}{dx_j} \frac{dv_i}{dx_j}$$

Also, it holds:

$$\int_\Omega \nabla \boldsymbol{w}^{n+1} : \mu_i \nabla^S \boldsymbol{u}^{n+1} \, \mathrm{d}\Omega = \int_\Omega \nabla^S \boldsymbol{w}^{n+1} : \mu_i \nabla^S \boldsymbol{u}^{n+1} \, \mathrm{d}\Omega$$

Applying the Green-Gauss divergence theorem, the following equality holds:

$$\int_\Omega \nabla \cdot \left(\boldsymbol{w}^{n+1} \cdot \mu_i \nabla^S \boldsymbol{u}^{n+1} \right) \, \mathrm{d}\Omega = \int_\Sigma \left(\boldsymbol{w}^{n+1} \cdot \mu_i \nabla^S \boldsymbol{u}^{n+1} \right) \cdot \boldsymbol{n}_\Sigma \, \mathrm{d}\Sigma = 0$$

with Σ being the domain's boundary and $\boldsymbol{n}_\Sigma$ its outward unit normal. By definition of $\boldsymbol{w}$, this part becomes zero.

Secondly, the pressure gradient is also integrated by parts and the divergence theorem is applied, again:

$$\int_\Omega \boldsymbol{w}^{n+1} \cdot \nabla p^{n+1} \, \mathrm{d}\Omega = \int_\Omega \nabla \cdot (\boldsymbol{w}^{n+1} p^{n+1}) \, \mathrm{d}\Omega - \int_\Omega p^{n+1} \nabla \cdot \boldsymbol{w}^{n+1} \, \mathrm{d}\Omega = \int_\Sigma \boldsymbol{w}^{n+1} p^{n+1} \boldsymbol{n}_\Sigma \, \mathrm{d}\Sigma - \int_\Omega p^{n+1} \nabla \cdot \boldsymbol{w}^{n+1} \, \mathrm{d}\Omega$$

in which the integral along the boundary of the domain is zero.

The same testing and integration is applied to the incompressibility condition, where the test function q satisfies $\{q \in \mathcal{L}^2(\Omega)\}$.

Therefore the weak form of the problem above can be summed up like: For given physical parameters gravity $\boldsymbol{g}$, surface tension coefficient τ and the density and viscosity fields ρ_i and μ_i, find the velocity $\boldsymbol{u}^{n+1}$ and pressure p^{n+1}, such that for all velocity test functions $\boldsymbol{w}$ and pressure test functions q:

$$\begin{aligned} &\frac{1}{\Delta t} \int_\Omega \boldsymbol{w}^{n+1} \cdot \rho_i \boldsymbol{u}^{n+1} \, \mathrm{d}\Omega + \int_\Omega \boldsymbol{w}^{n+1} \boldsymbol{u}^n \cdot \nabla \boldsymbol{u}^{n+1} \, \mathrm{d}\Omega \\ &+ \int_\Omega \nabla^S \boldsymbol{w}^{n+1} : \mu_i \nabla^S \boldsymbol{u}^{n+1} \, \mathrm{d}\Omega - \int_\Omega \nabla \boldsymbol{w}^{n+1} \cdot p^{n+1} \, \mathrm{d}\Omega = \\ &\frac{1}{\Delta t} \int_\Omega \boldsymbol{w}^{n+1} \rho_i \boldsymbol{u}^n \, \mathrm{d}\Omega + \int_\Omega \boldsymbol{w}^{n+1} \cdot \rho_i \boldsymbol{g} \, \mathrm{d}\Omega + \int_\Gamma \boldsymbol{w}^{n+1} \cdot \tau \kappa \boldsymbol{n}_\Gamma \, \mathrm{d}\Gamma \end{aligned} \tag{3.2a}$$

and

$$\int_{\Omega} \nabla \cdot q^{n+1} \cdot \boldsymbol{u}^{n+1} = 0 \tag{3.2b}$$

Remark that the convection term $\int_{\Omega} \boldsymbol{w}^{n+1} \boldsymbol{u}^{n} \cdot \nabla \boldsymbol{u}^{n+1} \, \mathrm{d}\Omega$ and the mass matrix $\frac{1}{\Delta t} \int_{\Omega} \boldsymbol{w}^{n+1} \rho_i \boldsymbol{u}^{n} \, \mathrm{d}\Omega$ contain information to both time steps, n and $n+1$ and therefore need a special treatment.

3.3 Stabilization of the convection Term

The finite element methods for fluids are known to require stabilization of the solution for cases with dominating convection, i.e., for flows with high Reynolds numbers. In this section the stabilization method we use is explained shortly.

The velocity stabilization is based on the method described in (Donea and Huerta, 2003) for incompressible Navier-Stokes flows. Here we have to adapt it to an enriched finite element formulation. Also, by programming issues, we simplified the formulation of the residual function in the stabilization term.

In our semi-implicit method, the stabilization adds another matrix to be multiplied with the unknown velocity $\boldsymbol{u}^{n+1}$. As in the convection term, we use the velocity of the time step before - time step n - to calculate the stabilization.

Following this both, the weak form of the stabilization matrix reads like:

$$\mathbf{M}_{\text{Stab}} = \sum_{e=1}^{n_{el}} \int_{\Omega^e} \left((\boldsymbol{u}^{n} \cdot \nabla) \boldsymbol{w}^{n+1} \left(\tau_{\text{SUPG}} \mathcal{R}(\boldsymbol{u}^{n}) \right) \right) \, \mathrm{d}\Omega$$

where $\mathcal{R}$ is the convection term:

$$\mathcal{R}(\boldsymbol{u}^{n}) = \rho \left(\boldsymbol{u}^{n} \cdot \nabla \right)$$

The resulting matrix is added to the left-hand-side of the Navier-Stokes problem and therefore directly influences the solution of for $\boldsymbol{u}^{n+1}$.

The stabilization parameter τ_{SUPG} is calculated using the following definition:

$$\tau_{\text{SUPG}} = \left(\frac{1}{\tau_{S1}^r} + \frac{1}{\tau_{S2}^r} + \frac{1}{\tau_{S3}^r} \right)^{-1/r}$$

where

$$\tau_{S1} = \frac{||\mathbf{c}||}{||\mathbf{k}||}, \qquad \tau_{S2} = \frac{\Delta t}{2}, \qquad \tau_{S3} = \tau_{S1} Re.$$

with the elemental convection and viscosity matrices:

$$\mathbf{c} = \int_{\Omega^e} \boldsymbol{w}^{n+1} (\boldsymbol{u}^n \cdot \nabla) \boldsymbol{u}^n \, \mathrm{d}\Omega \qquad \mathbf{k} = \int_{\Omega^e} (\boldsymbol{u}^n \cdot \nabla) \, \boldsymbol{w}^{n+1} \cdot (\boldsymbol{u}^n \cdot \nabla) \, \boldsymbol{u}^n \mathrm{d}\Omega$$

and the elemental Reynolds number

$$Re = ||\boldsymbol{u}_e^n||^2 \, \frac{\rho}{\mu} \, \frac{||\mathbf{c}||}{||\mathbf{k}||}$$

In the actual code, we follow the suggestion of (Donea and Huerta, 2003) and choose the parameter $r = 1$. Also, for the split elements, we have to make an assumption on ρ and μ in the elemental Reynolds number. For this, we use a weighted average of the physical parameters, corresponding to the ratio of each domain to the full element size.

3.4 Two-phase flow with level-set technique

As discussed in Chapter 1.3, we choose the level set technique to track the position of the interface. This is a useful way to be able to define the physical parameters in Eulerian formulations for large geometrical changes of the phases. The level set

function is a time-dependent scalar field $\phi(\boldsymbol{x}, t)$, defined as:

$$\phi(\boldsymbol{x}, t) = \begin{cases} -d & \text{for } \boldsymbol{x} \text{ in } \Omega_1 \\ 0 & \text{for } \boldsymbol{x} \text{ on } \Gamma \\ d & \text{for } \boldsymbol{x} \text{ in } \Omega_2 \end{cases}$$

with d being the distance of the points in $\boldsymbol{x}$ from the interface points, which are collected in $\boldsymbol{y}_\Gamma$; so d is defined as:

$$d = \min\left(||\boldsymbol{x} - \boldsymbol{y}_\Gamma||\right)$$

The density of the two phases is then defined in terms of the level-set function, i.e.,

$$\rho(\boldsymbol{x}) = \begin{cases} \rho_1 & \text{for: } \phi > 0 \\ \rho_2 & \text{for: } \phi < 0 \end{cases}$$

and the same for the viscosity:

$$\mu(\boldsymbol{x}) = \begin{cases} \mu_1 & \text{for: } \phi > 0 \\ \mu_2 & \text{for: } \phi < 0 \end{cases}$$

As one can see, the zero-level-set defines the position of the interface. For the surface tension integrals along the interface, a line integration routine is applied.

The interface geometry evolves over time, so the level set has to represent this movement. The interface movement is assumed to correspond to the velocity of the flow particles close to it. So the motion of the interface is represented correctly, if the level set is moved with the velocity field of the flow. This leads to the following transport scheme for the level set:

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi = 0 \quad \text{in: } \Omega, t > 0 \tag{3.3}$$

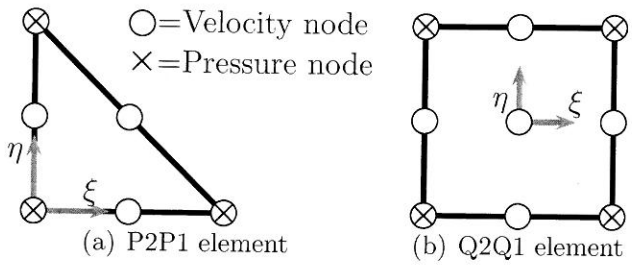


Figure 3.1: Schematics of the used P2P1 and Q2Q1 elements

In our numerical scheme, this convection step is performed on the end of each time step, therefore $\boldsymbol{u}^{n+1}$ is used to calculate the level set (and interface) for the following time-step.

3.5 Space discretization by X-FEM

The problem stated above is solved by the extended finite element method (X-FEM). With this, we are able to resolve the discontinuities of the the solution correctly. Especially the strong discontinuity of the pressure is represented directly by this enriched discretization. In fact, by choosing this, we are able to use the same mesh, irrespectively where the interface is situated in the current time step.

The solution is solved in terms of the discrete solution at a set of nodes in the triangulation of the domain Ω. We use the LBB-stable Q2Q1 and P2P1 elements, i.e., a quadratic interpolation of the velocity field and linear pressure discretization is used into each direction component. Figure 3.1 shows a schematic of the used elements.

Generally, in finite element methods, the discretized solution of the velocity and the pressure fields, u_h and p_h, are solved in terms of the nodal values and the elements' shape functions, for each time step

$$\boldsymbol{u}_h(\boldsymbol{x},t) = \sum_{j\in\mathcal{N}^{\boldsymbol{u}}} N_j^{\boldsymbol{u}}(\boldsymbol{x})\boldsymbol{u}_j(t) \qquad p_h(\boldsymbol{x},t) = \sum_{j\in\mathcal{N}^{p}} N_j^{p}(\boldsymbol{x})p_j(t)$$

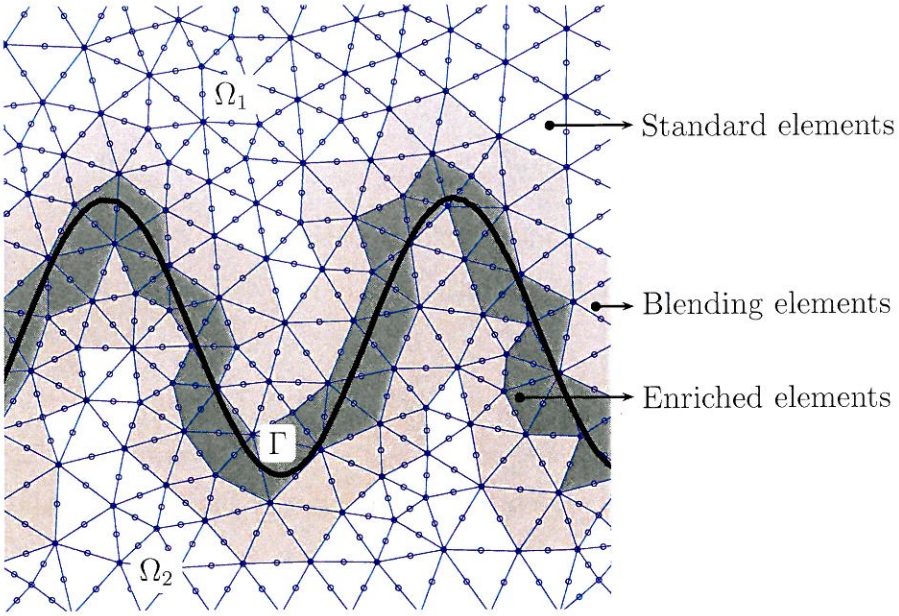


Figure 3.2: cos-example for element types: standard, blending and enriched elements

where $\mathcal{N}^{\boldsymbol{u}}$ and $\mathcal{N}^{p}$ denote the collection of all velocity and pressure nodes. $N^{\boldsymbol{u}}$ and N^{p} stand for the velocity and pressure shape functions.

Since the shape functions normally are continuous and smooth inside each element, this representation would lack accuracy for discontinuous solutions. In the present case, discontinuities are introduced due to the interface conditions. On the one hand, the velocity derivative becomes discontinuous due to the changing viscosities μ_1 and μ_2 of the two fluids. On the other hand, the surface tension of the interface introduces a localized force into the solution, which causes a discontinuous pressure.

So it becomes necessary to use a discretization which is able to resolve discontinuities in the elements, which are cut by the interface. As the interface position is tracked by the zero level of ϕ, the nodes of the elements containing this zero-level are collected in $\mathcal{N}_{enr}$, those which are not effected can use the standard finite element discretization mentioned above. A special treatment is needed for the nodes, which participate to both. Those elements are partly enriched and called blending elements. Figure 3.2 shows the entities of split, blending and standard elements.

3.5.1 Velocity enrichment - Ridge function

For representing the discontinuous velocity derivative on the interface, the so-called Ridge function is added to the discretization of the enriched elements, following (Moës et al., 2003):

$$R(\boldsymbol{x}) = \sum_{j \in \mathcal{N}_{enr}} |\phi_j| N_j(\boldsymbol{x}) - \left| \sum_{j \in \mathcal{N}_{enr}} \phi_j N_j(\boldsymbol{x}) \right|$$

and the discrete velocities, which is a summation of the standard and enriched part:

$$\boldsymbol{u}_h(\boldsymbol{x}, t) = \sum_{j \in \mathcal{N}} \boldsymbol{u}_j(t) N_j^{\boldsymbol{u}}(\boldsymbol{x}) + \sum_{j \in \mathcal{N}_{enr}} \boldsymbol{a}_j(t) M_j^{u}(\boldsymbol{x})$$

in which a_j are additional degrees of freedom representing the velocity-discontinuities in the elements. Its interpolation function $M^{\boldsymbol{u}}$ is defined as

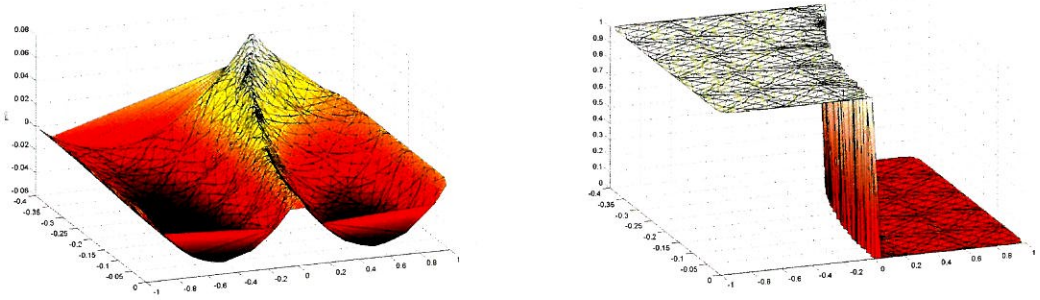
$$M_j^{\boldsymbol{u}}(\boldsymbol{x}) = R(\boldsymbol{x}) N_{\mathrm{enr},j}^{\boldsymbol{u}}(\boldsymbol{x})$$

In general, the shape functions applied to the enrichment functions ($N_{\mathrm{enr},j}^{\boldsymbol{u}}(\boldsymbol{x})$) may be different than those for the standard part. But here, we always applied the same shape functions for the standard and the enriched part.

In Figure (3.3(a)) such a two-dimensional Ridge enrichment function is shown for a quadrilateral element, which is cut by an interface. Note the continuous, but non-smooth properties of the enrichment function at the interface.

3.5.2 Pressure enrichment - Heaviside function

In contrast to the velocity, the pressure is discontinuous along the interface. The surface tension introduces localized forcing terms along the interface. Therefore, we



(a) 2D Ridge function

(b) 2D Heaviside funcion

Figure 3.3: Pressure and velocity enrichment functions on a split element. The velocity enrichment leads to a non-smooth solution, the pressure enrichment to a discontinuous solution

choose the pressure discretization to be enriched by the Heaviside function, which is a step function with the step located at the interface:

$$\mathrm{H}(\phi(\boldsymbol{x})) = \begin{cases} 1 & \text{for } \phi > 0 \\ 0 & \text{for } \phi <= 0 \end{cases}$$

and the pressure discretization from the shape functions reads then like

$$p_h(\boldsymbol{x}, t) = \sum_{j \in \mathcal{N}} p_j(t) N_j^p(\boldsymbol{x}) + \sum_{j \in \mathcal{N}_{enr}} b_j(t) N_j^p(\boldsymbol{x}) \mathrm{H}(\phi(\boldsymbol{x}))$$

where $b_j(t)$ represents the additional pressure unknowns due to the enrichment. With $b \neq 0$, the discretization of the pressure field becomes discontinuous.

In Figure (3.3(b)) such a Heaviside enrichment function is shown for a quadrilateral element. Remark the discontinuity of the enrichment function at the interface.

3.6 Summing up: Unknowns and matrices

In the numerical code, the problem is translated into the matrix formulation as following:

$$\begin{cases} \frac{1}{\Delta t}\mathbf{M}\mathbf{u}^{n+1} + \mathbf{C}\mathbf{u}^{n+1} + \mathbf{K}\mathbf{u}^{n+1} + \mathbf{M}_{\text{stab}}\mathbf{u}^{n+1} + \mathbf{G}^{\text{T}}\mathbf{p}^{n+1} & = \frac{1}{\Delta t}\mathbf{M}\mathbf{u}^{n} + \mathbf{f}_g + \mathbf{f}_\Gamma \\ \mathbf{G}\mathbf{u}^{n+1} & = \mathbf{0} \end{cases} \tag{3.4}$$

plus sufficient boundary conditions. This can be transformed into one system of equations, which needs to be solved. Here, the solution for the velocity and the pressure are solved in one step, so it is a velocity-pressure coupled solution:

$$\begin{bmatrix} (\frac{1}{\Delta t}\mathbf{M} + \mathbf{C} + \mathbf{K} + \mathbf{M}_{\text{stab}}) & \mathbf{G}^{\text{T}} \\ \mathbf{G} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ \mathbf{p}^{n+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t}\mathbf{M}\mathbf{u}^{n} + \mathbf{f}_g + \mathbf{f}_\Gamma \\ \mathbf{0} \end{bmatrix}$$

plus the additional equations coming from the Lagrange multipliers for the Dirichlet boundary conditions.

The physical meaning of the parts of the equation above can be described like following:

- $\mathbf{M}\mathbf{u}^{n+1}$ represents the mass inertia of the two fluids, it ensures the conservation of the inertia of the fluids.
- $\mathbf{C}\mathbf{u}^{n+1}$ introduces the effect of the convection into the system of equation. Again, it has to be pointed out that $\mathbf{C}$ is calculated dependent on the velocity of the previous time-step.
- $\mathbf{K}\mathbf{u}^{n+1}$ stands for the deviatoric stresses inside the flow and is controlled by the viscosity of each the materials.
- $\mathbf{M}_{\text{stab}}\boldsymbol{u}^{n+1}$ is the stabilization matrix. It has no real physical meaning, but is needed to stabilize convection-dominated flows in finite elements.

- $\mathbf{G}\mathbf{p}^{n+1}$ is the pressure gradient and introduces the flows' acceleration from high pressure to low pressure. Also, in the second equation, $\mathbf{G}$ is used to ensure the incompressibility condition.
- $\mathbf{f}_g$ is the gravitational forcing vector
- $\mathbf{f}_\Gamma$ is the vector which introduces the forces by the interface.

For solving the full system, the unknown vectors $\mathbf{u}$ and $\mathbf{p}$ are organized as:

$$\mathbf{u} = \begin{bmatrix} u_1^x & u_1^y & u_2^x & u_2^y & \dots & u_n^x & u_n^y & a_1^x & a_1^y & \dots & a_{\bar{n}}^x & a_{\bar{n}}^y \end{bmatrix}^{\mathrm{T}}$$

and

$$\mathbf{p} = \begin{bmatrix} p_1 & p_2 & \dots & p_m & b_1 & \dots & b_{\hat{m}} \end{bmatrix}^{\mathrm{T}}$$

with u_n^x and u_n^y being the velocity components for the nth velocity node and $a_{\bar{n}}^x$ and $a_{\bar{n}}^y$ the enriched velocity for the $\bar{n}$th enriched nodes. The same counts for the pressure: p_m is the pressure at the mth pressure node and $b_{\hat{m}}$ the pressure enrichment degree of freedom. For calculating the matrices, these vectors are reshaped as following:

$$\mathbf{u} = \begin{bmatrix} u_1^x & u_1^y \\ u_2^x & u_2^y \\ \vdots & \vdots \\ u_n^x & u_n^y \\ a_1^x & a_1^y \\ a_2^x & a_2^y \\ \vdots & \vdots \\ a_{\bar{n}}^x & a_{\bar{n}}^y \end{bmatrix}$$

in multiplication with the shape functions vector, the local velocity can be determined like:

$$\boldsymbol{u}_h = \mathbf{N}^{\boldsymbol{u}}\mathbf{u}$$

with the shape function vector

$$\mathbf{N}^{\boldsymbol{u}} = \left[\begin{array}{cccccc} N_1^{\boldsymbol{u}} & N_2^{\boldsymbol{u}} & \dots & N_n^{\boldsymbol{u}} & M_1^{\boldsymbol{u}} & \dots & M_{\bar{n}}^{\boldsymbol{u}} \end{array} \right]$$

with $N^{\boldsymbol{u}}$ being the biquadratic velocity shape functions and $M_1^{\boldsymbol{u}}$ being the enrichment function, resulting from multiplication of the Ridge function and the enrichment shape functions for each of the enriched velocity node.

The same for the pressure:

$$p_h = \mathbf{N}^p \mathbf{p}$$

with the shape function vector

$$\mathbf{N}^p = \left[\begin{array}{cccccc} N_1^p & N_2^p & \dots & N_n^p & M_1^p & \dots & M_{\bar{n}}^p \end{array} \right]$$

with N^p being the bilinear pressure shape functions and M^p being the enrichment function, resulting from the Heaviside function multiplied with the shape functions for each of the enriched pressure node. Also, for some terms, a second form of the shape function vector is needed, taking the two-dimensional formulation into account:

$$\mathbf{N}^{\boldsymbol{u}*} = \left[\begin{array}{cccccccccccc} N_1^{\boldsymbol{u}} & 0 & N_2^{\boldsymbol{u}} & 0 & \dots & N_n^{\boldsymbol{u}} & 0 & M_1^{\boldsymbol{u}} & 0 & \dots & M_{\bar{n}}^{\boldsymbol{u}} & 0 \\ 0 & N_1^{\boldsymbol{u}} & 0 & N_2^{\boldsymbol{u}} & \dots & 0 & N_n^{\boldsymbol{u}} & 0 & M_1^{\boldsymbol{u}} & \dots & 0 & M_{\bar{n}}^{\boldsymbol{u}} \end{array} \right]$$

It has to be noted, that for standard (not-enriched) elements, the velocity and pressure vectors, as well as the shape function vectors are just those for standard finite elements, i.e., without the enrichment degrees of freedom and shape functions.

For the transient Navier-Stokes problem stated above, the following matrices are involved:

- The mass matrix:

$$\mathbf{M} = \int_{\Omega} \mathbf{N}^{\boldsymbol{u}*\mathrm{T}} \rho_i \mathbf{N}^{\boldsymbol{u}*} \, \mathrm{d}\Omega$$

- The convection matrix:

$$\mathbf{C} = \int_{\Omega} \mathbf{N}^{\boldsymbol{u}*\mathrm{T}} \rho_i \mathbf{u}^n \cdot \nabla \mathbf{N}^{\boldsymbol{u}} \, \mathrm{d}\Omega$$

- The viscosity matrix:

$$\mathbf{K} = \int_{\Omega} \mathbf{B}^{\mathrm{T}} \mathbf{I} \mathbf{B} \, \mathrm{d}\Omega$$

with $\mathbf{B}$ being the stiffness matrix and $\mathbf{I}$ describes the physical stiffness parameters. They slightly differ between the 2D and the axis symmetric formulation: For the two-dimensional cases, $\mathbf{B}$ reads like:

$$\mathbf{B} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \mathbf{N}^{\boldsymbol{u}*}$$

and $\mathbf{I}$ is defined as $\mu_i \cdot \mathrm{diag}\,(2, 2, 1)$. Secondly, for the axis symmetric problem, the following formulation for $\mathbf{B}$ holds:

$$\mathbf{B} = \begin{bmatrix} \frac{\partial}{\partial r} & 0 \\ 0 & \frac{\partial}{\partial z} \\ \frac{1}{r} & 0 \\ \frac{\partial}{\partial z} & \frac{\partial}{\partial r} \end{bmatrix} \mathbf{N}^{\boldsymbol{u}*}$$

and $\mathbf{I}$ is defined as $\mu_i \cdot \mathrm{diag}\,(2, 2, 2, 1)$.

- The pressure gradient:

$$\mathbf{G} = -\int_{\Omega} \nabla \mathbf{N}^{p\mathrm{T}} \mathbf{N}^{\boldsymbol{u}} \, \mathrm{d}\Omega$$

- The gravity forcing term

$$\mathbf{f}_g = \int_{\Omega} \mathbf{N}^{\boldsymbol{u}*} \mathbf{g} \, \mathrm{d}\Omega$$

- The interface forcing term

$$\mathbf{f}_\Gamma = \int_\Gamma \mathbf{N}^{u*} \tau \kappa \mathbf{n} \, \mathrm{d}\Gamma$$

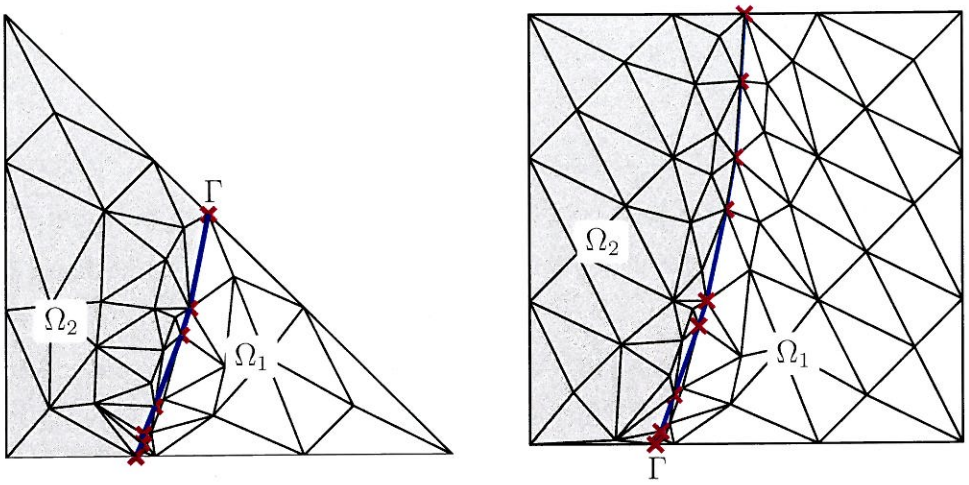
The stabilization forcing term is defined in Section 3.3.

3.7 Enriched integration with triangular integration cells

Due to the discontinuities of the physical parameters, a standard integration rule is not accurate enough for the domain integration of the elements which are intersected by the interface. We choose to perform the integration of these split elements with the help of triangular integration cells. On the one hand, this enables us to integrate correctly for each of the subdomains in the split elements. Secondly, the points of the integration cells are also used to localize the interface position inside each element. This interface position is then used for the line integration of the surface tension force.

Figure 3.4 shows the triangular integration cells for a quadrilateral and for a triangular mesh element. Initially, we position the integration cells regularly inside each split element. The cells are then used to find the exact position of the interface inside each element. Finally, the connectivity of the integration cells is changed such that none of it is split by the interface.

Since the reference elements are organized such that none of them is intersected by the interface, a standard integration rule may be used for each of the reference cells. The Gauss points are positioned with respect to the triangular reference cells, and the Gauss point weights are multiplied with the area ratio of the reference triangle and the full element.



(a) Tri element (b) Quad element

Figure 3.4: Reference mesh for enriched elements integration; the reference mesh is initialized with 3 triangles per element side and then modified along the interface

Additionally, the points of the reference mesh give a detailed representation of the interface. These points are also used for the interface line integration, which is described in Section 3.8.

For the upcoming calculations of the examples, the integration of the standard (non-split) elements is done with a 9-point Gauss integration for Q2Q1 elements and a 7-point integration for the P2P1 elements. Secondly, after dividing the split elements into the integration cells, we apply a 16-points Gauss integration for each of the triangular integration cells.

3.8 Calculation of the surface tension term

The surface tension comes into the calculation as an additional forcing term. As already mentioned, it is defined as

$$\sigma = \tau \delta_\Gamma \kappa \mathbf{n}$$

or, in the weak form after spatial discretization and resolving the δ-Function

$$\mathbf{f} = \tau \int_{\Gamma} \mathbf{N}^{\boldsymbol{u}} \kappa \mathbf{n} \, \mathrm{d}\Gamma$$

with τ being the surface tension coefficient – which is a physical parameter – and κ being the curvature of the interface and $\mathbf{n}$ being the interface normal direction.

Remark that this integral is an line integral and it needs to be computed separately. By choosing this integration, the surface tension is introduced explicitly on the interface. This stands in contrast to most other implementations, were the surface tension term is introduced in the form of a domain integral and localized by a delta function $\delta(\phi(x))$, dependent on the distance from the interface, which is the level set function (see e.g., (Chessa and Belytschko, 2003), (Marchandise et al., 2007), (de Sousa et al., 2004) and (Tornberg and Enquist, 2000)).

We choose to do so, because the combination of the $\delta(\phi(x))$-function together with domain integration has higher demands on the level set function: it needs to stay a distance function. In our method, we are just interested into the zero-level-set for the localization of the interface, and therefore ϕ may differ from a distance function. So we need to re-initialize ϕ less often compared to the domain-integration.

Also, oscillations of the level set which are not directly on the interface, but a bit beside, may still influence the calculation of the surface tension. Since we only use information about the zero level set, those oscillations do not cause problems for our method.

As mentioned, the location of the interface is defined by the zero-level-set, which is determined during calculating the reference integration mesh for the split elements. The resulting interface segments have to be sorted such that for each interface segment, the neighbors are known. This sorting requires some programming effort. Once sorted, the interface can be parametrized by an arc length parameter t, so we result in the two

functions $x_\Gamma(t)$ and $y_\Gamma(t)$, which describe the interface position. This parametrized interface is then used to calculate the curvature and the normal direction.

The interface curvature is defined like in (Hermann and Klette, 2007):

$$\kappa(t) = \frac{\begin{vmatrix} x' & y' \\ x'' & y'' \end{vmatrix}}{(x'^2 + y'^2)^{\frac{3}{2}}}$$

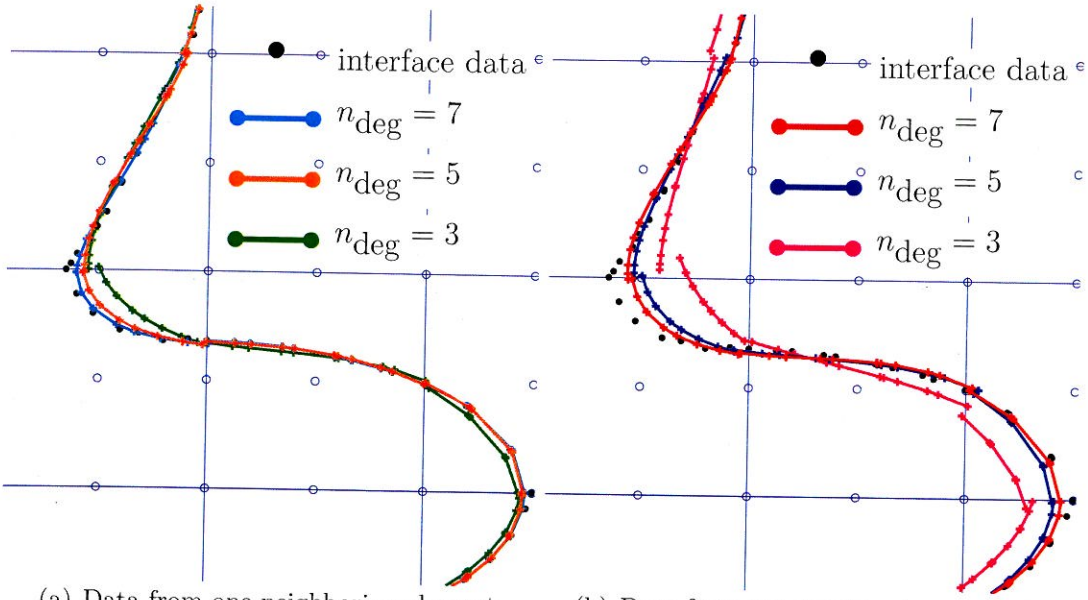
with the definition for the derivatives $x' = \frac{\partial x(t)}{\partial t}$ and $x'' = \frac{\partial^2 x(t)}{\partial t^2}$.

The derivatives of the parametrized interface are also used to find the normal direction vector $\mathbf{n}$:

$$\mathbf{n} = \begin{pmatrix} -y' \\ x' \end{pmatrix}$$

We choose to calculate the derivatives in terms of a piecewise polynomial approximation of the parametrized interface. For the element of interest, the interface points of the n_{Neigh} neighboring elements are used as input for a polygonal least-squares. The actual polynomial approximation is done by the MATLAB build-in function **polyfit**, which directly results the coefficients of the interface approximation polynomial. With the coefficients, the values of the derivatives can be easily computed.

In this approximation, we have two parameters, which influence the approximation of the interface: The degree of the polynomial and the number of neighbors taken into consideration for the approximation of the interface segment of each element. The higher the degree of the polynomial, the better the approximation is able to follow the interface data. But for very high degrees oscillations may appear. Additionally, the number of data points limits the degree of the polynomial. Especially for elements on the boundary or very few reference elements, there may not be enough points for a complete description of a high-order polynomial.



(a) Data from one neighboring element (b) Data from two neighboring elements

Figure 3.5: Approximation of the interface as polynomials: Different polynomial degrees and number of neighbors

Secondly, the number of neighbors influences the approximation. As for each element the polynomial changes due to the change of the data points, strong discontinuities of the interface position appear at the element edges. Therefore also the curvature and the normal directions are not continuous at these element edges. Since the data change is generally smaller for a higher number of neighbors taken into account, one could expect that the discontinuities become less the more neighbors are used. Tests for this have shown, that this effect is not that strong. In contrast, the least squares fitting works better if we only take one neighbor into consideration. This may be due to the way, the least-square fitting in **polyfit** works: It minimizes the over-all error of the polynomial to *all* data points, including the points of the neighbors, which may be far away from the points of interest. Figure 3.5 illustrates this. Finally we choose polynomials of order 7 for the interface approximation. The data points for each element are from the element itself and from one neighboring element to each side.

For the integration of the surface tension forcing term, we use a standard one-

dimensional Gauss quadrature rule between each of the data points segments.

Curvature in axisymmetric cases

In the axisymmetric case, the curvature needs some extra treatment. Following (Chessa and Belytschko, 2003), to the in-plane curvature, determined by the polynomial approximation, the rotational curvature has to be added:

$$\kappa_{\text{axi}} = \kappa + \kappa_{\text{rot}}$$

and this is defined as:

$$\kappa_{\text{rot}} = \frac{\mathbf{n}_r}{r}$$

where $\mathbf{n}_r$ is the radial component of the interface normal and r the radial coordinate. Also, to avoid infinite curvature of the interface close to the axis of rotation, a symmetric condition of the interface has to be added. In the present code, we choose to introduce this symmetry only in the polynomial approximation. This is done by mirroring the data points for the elements close to the interface to the negative r-coordinate side. By this, the polynomial is always symmetric to the $r = 0$-axis and the curvature and normals can be computed from this.

3.9 Projection of the velocity for different discretization along time

One of the crucial points about enriched finite elements is, that due to the enrichment, the interpolation space of the variables, here namely the velocity $\boldsymbol{u}$ and pressure p changes as their enrichment changes. The enrichment functions are dependent on the level set function, which is updated on every time step.

In the system solved in the present code, we are dealing with the velocity of the known time-step n and solving for the velocity of time-step $n+1$, with an individual discretization for both. In order to be able to solve the problem within one interpolation space, we decided to project the u_n-velocity to the discretization of time-step $n+1$, using a least-square fitting.

So we have to project the solution of step n

$$\boldsymbol{u}^n = \sum_i N_i^n u_i + \sum_j \hat{N}_j^n u_j^{\text{enr}}$$

into the space with the shape functions of time-step $n+1$:

$$\boldsymbol{U}^{n+1} = \sum_i N_i^{n+1} a_i + \sum_k \hat{N}_k^{n+1} a_k^{\text{enr}}$$

For first, the standard part of the discretization does not change between the time steps. So only the enriched part of $\boldsymbol{u}^n$ needs to be projected, and their contribution will be added to the standard part. Thus our problem reads like finding the best approximation

$$\boldsymbol{V}^n = \sum_i N_i a_i + \sum_k \hat{N}_k^{n+1} a_k^{\text{enr}}$$

that fits best the data given from

$$\boldsymbol{u}^n = \sum_j \hat{N}_j^n u_j^{\text{enr}}$$

The system for solving this in a least-square approach can be formulated like

$$\mathbf{Ma} = \mathbf{f}$$

with

$$\mathbf{M} = \left[\begin{array}{c|c} \mathbf{M}^{11} & \mathbf{M}^{12} \\ \hline \mathbf{M}^{21} & \mathbf{M}^{22} \end{array}\right] \qquad \begin{cases} M_{ij}^{11} = \int_\Omega N_i N_j \, \mathrm{d}\Omega \\ M_{ij}^{12} = \int_\Omega N_i \hat{N}_j^{n+1} \, \mathrm{d}\Omega \\ M_{ij}^{22} = \int_\Omega \hat{N}_i^{n+1} N_j^{n+1} \, \mathrm{d}\Omega \end{cases}$$

the right hand side vector

$$\mathbf{f} = \left\{\begin{array}{c} \mathbf{f}^1 \\ \hline \mathbf{f}^2 \end{array}\right\} \qquad \begin{cases} f_i^1 = \int_\Omega N_i u^n \, \mathrm{d}\Omega \\ f_i^2 = \int_\Omega \hat{N}_i^{n+1} u^n \, \mathrm{d}\Omega \end{cases}$$

and the vector of unknowns

$$\mathbf{a} = \left\{\begin{array}{c} \mathbf{a}^1 \\ \hline \mathbf{a}^2 \end{array}\right\} \qquad \begin{cases} a_i^1 = a_i \\ a_i^2 = a_i^{enr} \end{cases}$$

The projected velocity is then calculated by adding the standard part of the nth time step and the standard part of $\mathbf{a}$. The new enriched part is then $\mathbf{a_2}$.

3.10 Time and space discretization of the level set

3.10.1 Time discretization

As already mentioned above, the level set function evolves over time, the convection equation from Equation (3.3) is applied for this.

As time stepping scheme, we choose the two-step third-order Taylor-Galerkin (2S-TG3), as described in (Donea and Huerta, 2003) and already used for the movement of a multiphase level set in (Zlotnik, 2008). For the level-set, the following formulation

is used

$$\begin{aligned} \tilde{\phi^n} &= \phi^n + \frac{1}{3}\Delta t\dot{\phi^n} + \alpha\Delta t^2\ddot{\phi^n} \\ \phi^{n+1} &= \phi^n + \Delta t\dot{\phi^n} + \frac{1}{2}\Delta t^2\ddot{\tilde{\phi}^n} \end{aligned}$$

with $\dot{(\cdot)}$ and $\ddot{(\cdot)}$ being the first and second time derivatives. The parameter α is set to 1/9, following the recommendation of (Donea and Huerta, 2003).

As mentioned in (Zlotnik, 2008), the second time derivative of the level-set can be formulated as:

$$\ddot{\phi} = -\boldsymbol{u}\cdot\nabla\dot{\phi} = \boldsymbol{u}^{n+1}\cdot\nabla\left(\boldsymbol{u}^{n+1}\cdot\dot{\phi}\right) = \nabla\cdot\left(\left(\boldsymbol{u}^{n+1}\cdot\boldsymbol{u}^{n+1}\right)\nabla\phi\right)$$

thus the second part of the 2S-TG3 algorithm reads like:

$$\tilde{\phi^n} = \phi^n + \frac{1}{3}\Delta t\left(-\boldsymbol{u}^{n+1}\cdot\nabla\phi^n\right) + \alpha\Delta t^2\nabla\cdot\left(\left(\boldsymbol{u}^{n+1}\cdot\boldsymbol{u}^{n+1}\right)\nabla\phi^n\right) \tag{3.5}$$

Since this scheme is an explicit scheme, it constraints the time step size by its stability condition. This is given in form of the *Courant* vectors, which are for two dimensional cases defined like: $c_x = u_x\Delta t/h_x$ and $c_y = u_y\Delta t/h_y$ with h_x and h_y standing for the elemental sizes. The stable time step size can be chosen such that it fulfills the following condition:

$$c_x^2 + c_y^2 = 3/4\cdot\theta \tag{3.6}$$

which means, that for choosing $\theta = 0.9$ the particles travel approximately nine tens of the smallest element size of the mesh. (Zlotnik, 2008). In the practical implementation, this is the time step size we choose normally for the time integration for the full system; but for higher accuracy and stability, the actual value of θ may also be reduced.

Since cell size and velocity may differ in the full domain, a key point for the stability of the transport of the level set is the approximation of the cell size. For this, in the code an assumption is made based on the full size of the domain and the total number of elements. Since, this is not accurate and very high refinements in specific areas require smaller time steps, the value of θ is normally chosen relatively small.

Additionally, to gain higher stability, we normally perform one time step piecewise. For this, one time step is divided into a certain number of smaller intermediate time steps, for which the same velocity field is applied.

3.10.2 Spatial discretization

In space, the level set is discretized using the same shape functions as the velocity, i.e., the biquadratic shape functions, but without enrichment:

$$\phi(\boldsymbol{x}, t) \simeq \phi^h = \sum_{i \in \mathcal{N}} N_i(\boldsymbol{x}) \phi_i(t) = \mathbf{N}^{\boldsymbol{u}} \boldsymbol{\phi}$$

with

$$\boldsymbol{\phi} = [\phi_1, \phi_2, \ldots, \phi_{i_\mathcal{N}}]^T$$

After discretization the transport equation of the level-set reads like:

$$\mathbf{M}_\phi \dot{\boldsymbol{\phi}} + \mathbf{G}_\phi \phi = \mathbf{0}$$

where

$$\mathbf{M}_\phi = \int_\Omega \mathbf{N}^{\boldsymbol{u}T} \mathbf{N}^{\boldsymbol{u}} \, \mathrm{d}\Omega$$

and

$$\mathbf{G}_\phi = - \int_\Omega \mathbf{N}^{\boldsymbol{u}T} \boldsymbol{u}^T \left(\nabla \mathbf{N}^{\boldsymbol{u}} \right) \, \mathrm{d}\Omega$$

So the TG3-2S time stepping scheme is the following:

$$\mathbf{M}_\phi \tilde{\boldsymbol{\phi}}^{n+1} = \left[\mathbf{M}_\phi + \frac{1}{3}\Delta t \mathbf{G}_\phi + \alpha \Delta t^2 \mathbf{K}_\phi\right] \boldsymbol{\phi}^n \tag{3.7}$$

$$\mathbf{M}_\phi \boldsymbol{\phi}^{n+1} = \left[\mathbf{M}_\phi + \Delta t \mathbf{G}_\phi\right] \boldsymbol{\phi}^n + \frac{1}{2}\Delta t^2 \mathbf{K}_\phi \tilde{\boldsymbol{\phi}}^{n+1} \tag{3.8}$$

where $\mathbf{K}_\phi$ comes from the discretization of the last term of Equation (3.5):

$$\mathbf{K}_\phi = -\int_\Omega (\boldsymbol{u} \cdot \boldsymbol{u})(\nabla \mathbf{N}^{\boldsymbol{u}T} \nabla \mathbf{N}^{\boldsymbol{u}}) \, \mathrm{d}\Omega$$

So for the intermediate time steps, we solve the Equations (3.8) and (3.8) a certain number of times, but keeping the matrices $\mathbf{M}$, $\mathbf{G}$ and $\mathbf{K}$ the same.

3.11 Level set maintenance: reinitialization and smoothing

Over time, as the level set is transported corresponding to the flow velocities, it looses its property of being a signed distance function. In contrast to the methods which use the gradient of the level set to find the the interface normal direction and curvature, (as in (Chessa and Belytschko, 2003) and in (Marchandise et al., 2007)), we do not rely on this property due to the polynomial approximation of the interface.

Nevertheless, maintenance of the level set is required to avoid distortions. Moreover, due to inaccuracies and by spurious currents close to the interface, the interface becomes "noisy". So we included a reinitialization procedure into the code as well as a smoothing of the level set.

3.11.1 Reinitialization of the level set

The reinitialization of the level set function takes place after a certain number of time steps. In this, the signed-distance property of the level set is rebuilt. For this, for each node the distance to each interface point is calculated in a brute force manner, and the closest distance is applied to the node. Compared to the integration this does not take too long in our two-dimensional cases.

By this, the level set function becomes a signed distance function, but the position of the level set may change slightly and the mass conservation can not be guaranteed. To circumvent this, others proposed to exclude the nodes of the split elements from the reinitialization, such that the actual position of the interface stays the same. But still, this does not solve the problem of very steep level set gradients in the split element, if the interface moves strictly into one direction.

3.11.2 Smoothing of the level set

Secondly, we use a method to smooth the level-set function. (Groß et al., 2006) as well as (Tornberg and Enquist, 2000) apply for this a method which can be compared to adding artificial diffusion to the level set. It reads like:

$$\phi_{\text{smooth}} - \epsilon \Delta \phi_{\text{smooth}} = \phi$$

with ϵ being a very small artificial smoothing parameter. So the following system has to be solved:

$$\left[\mathbf{M}_\phi + \epsilon \mathbf{K}_\phi\right] \boldsymbol{\phi}_{\text{smooth}} = \mathbf{M}_\phi \boldsymbol{\phi}$$

with $\mathbf{M}_\phi$ being the mass matrix and $\mathbf{K}_\phi$ the stiffness matrix of the level set. This function effectively smooths out the oscillations of the level set. But on the other hand it changes also the position of the interface and by this causes some problems with the mass conservation. For example, if there is a relatively small part of a subdomain

(e.g., a small bubble), the level set has a small peak in its environment. By smoothing too much, the information about the interface may be smoothed out over time. This smoothing procedure is applied at the end of the time-step, directly after the new level set is calculated.

As one can see, both methods of the level set maintenance directly influence the mass-conservation of the flow. But nevertheless, in problematic settings they may be needed for keeping the stability of the code. So for each individual problem, a good trade-off has to be found between stability and mass conservation while applying this techniques.

Chapter 4

Validation of the code

In the following chapter, the performance of the code is described for several examples. This is to show the accuracy of the code and its ability to predict correctly the physical phenomena regarding two-phase flows and surface tension. The examples are chosen to give a wide overview on the code abilities, but also its limitations.

The first case is an artificial example from (Li et al., 2007), in which prescribed interface forces produce a discontinuous pressure and a non-smooth velocity solution. For this, an analytical solution exists and we can use it to show the mesh-convergence of the code with and without enrichment.

The first physical example is a Stokes flow with a rotating source term, that is reversed after a certain time. We deactivated the density-dependent terms, the flow becomes reversible and therefore the initial state should be reached again, which makes inaccuracies obvious. This example is used to present the different parameters in the movement and maintenance of the level set. Also, a setting without reversed source term shows the effect of the surface tension in general.

Secondly, the code is applied to the Rayleigh-Taylor-instability. For this, we compare the results with numerical experiments performed in (Smolianski, 2001).

Another standard example for two-phase flows are rising bubbles. For this, validation with experiments are available. The results we gained are compared with both, calculations and experimental data, as it is performed in (Hua and Lou, 2007).

The level set method also enables us to solve problems with a large change of the topology of the interface. We apply the present method to a setting of two bubbles rising to a free surface. This includes the merging of bubbles and the break-up of separating layers. This example is performed similarly in (Tornberg and Enquist, 2000), which compare our results with.

4.1 Steady cases with analytical solution

The steady case in this section is used to show the ability of the code to represent the discontinuities correctly. The example is the third example of (Li et al., 2007) and is constructed such that the pressure is discontinuous and the velocity non-smooth due to the localized force terms on the interface. Since the exact solution is known, it is possible to calculate the global error of the solution. We will use the error calculation to show the effect of the enrichment on discontinuous and non-smooth solutions.

The solution is realized by discontinuous body forces and a prescribed interface force, which acts in normal and tangential direction along the interphase. For the calculations in this section, we choose the viscosities to be $\mu_1 = 0.1$ and $\mu_2 = 1$. The actual description of the interface forces can be found in (Li et al., 2007) and they result in the following exact solution:

The exact velocity solution reads like

$$\boldsymbol{u} = \begin{cases} \begin{bmatrix} \frac{y}{4} \\ -\frac{x}{4}(1-x^2) \end{bmatrix} & \text{for: } x^2+y^2 > 1 \\ \begin{bmatrix} \frac{y}{4}(x^2+y^2) \\ -\frac{xy^2}{4} \end{bmatrix} & \text{for: } x^2+y^2 \leq 1 \end{cases}$$

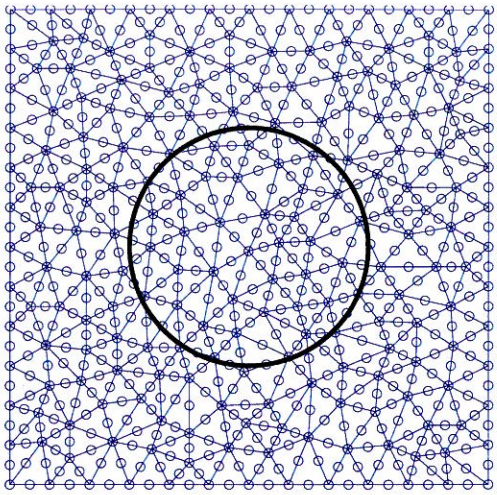


Figure 4.1: Mesh and interface position for the mesh $h = 0.2102$: In contrast to a regular mesh, the interface intersects the elements arbitrary to avoid regular effects of the elements sides being close to the interface.

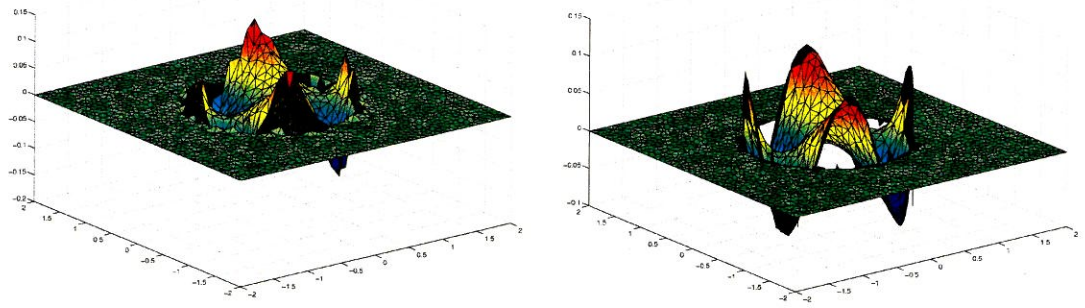
and the pressure

$$p = \begin{cases} \left(-\frac{3}{4}x^3 + \frac{3}{8}x\right) y & \text{for: } x^2 + y^2 > 1 \\ 0 & \text{for: } x^2 + y^2 \leq 1 \end{cases}$$

The problem is calculated on seven meshes with different levels of refinement. Here, we choose unstructured triangular meshes to minimize the problems of a regular mesh on the regular interphase. Such problems could appear, if the interphase is situated directly on element edges. In Figure (4.1) the mesh and the interface is shown for the relatively coarse mesh with $h = 0.2102$.

Figure (4.2) shows the effect of the pressure enrichment on the solution. Without enrichment, the discretization is not able to represent the jump of the pressure correctly. Also, in Table 4.1 the pressure and velocity errors are listed for the different meshes to show the mesh convergence.

In Figure (4.3) the velocity and pressure error over the average cell size is plotted in a log-log plot. In this, we can see, how the pressure and velocity enrichment improve



(a) Without enrichment (b) With Heaviside-enrichment

Figure 4.2: Effect of Heaviside-enrichment of the pressure for the non-physical example of interphase forces; $h = 0.071935$

Table 4.1: Mesh convergence on constructed steady case

h	Err$u$ not enr	Err$p$ not enr	Err$u$ enr	Err$p$ enr
0.308610	0.092320	0.077095	0.015388	0.039343
0.210240	0.065356	0.058607	0.005457	0.018059
0.146650	0.041116	0.040417	0.002225	0.017028
0.103760	0.033804	0.028989	0.001419	0.004671
0.071935	2.0089e-02	2.1853e-02	5.8163e-04	4.2861e-03
0.050751	1.3651e-02	1.3722e-02	4.2931e-04	2.0123e-03
0.035814	9.3204e-03	1.0344e-02	1.1152e-04	6.8339e-04

the solution. Overall the errors with enrichment are smaller, but we also observe a higher mesh convergence rate. For the calculations without enrichment, the results seem to be first order accurate, but with the enrichment of pressure and velocity, the convergence rates increase up to approximately two. This corresponds to what the "augmented approach" of (Li et al., 2007) reached in their investigations. This shows the importance of enriching the discretization for non-smooth or discontinuous problems. In physical problem, such discontinuities are brought into the solution of two-phase flows by the surface tension and the jump of the viscosities. Here we have shown, that by enrichment we can handle such discontinuities better.

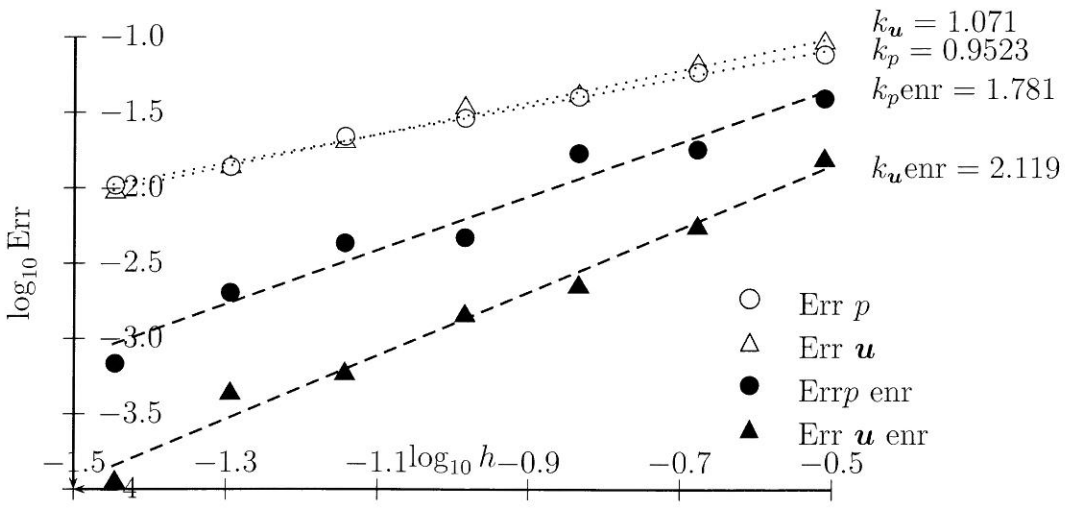


Figure 4.3: Mesh convergence for a steady case: log/log-plot of the velocity and pressure error over the cell size shows much better results with enrichment.The order of convergence k is of around 1 for the not enriched, whereas by enrichment the solution converges with the order of around two

4.2 Bubble in a rotating Stokes flow

In this section, the features and performance of the code regarding the time stepping of the level set are discussed in general. The effect of the most important parameters for this, the mesh size and the time step size is shown. But also, we show the impact of the level set maintenance (reinitialization and smoothing) on the mass conservation properties. In addition, this example is used to demonstrate the effect of the surface tension in general.

The example is described like the following: In a quadratic domain of the dimensions $[-2,2] \times [-2,2]$, a circulating flow is constructed. For the first time period, the flow is clockwise oriented, after that counter-clockwise. This is realized by the following source terms:

$$\mathbf{f}(\boldsymbol{x}) = \begin{cases} \begin{pmatrix} +x_2 \\ -x_1 \end{pmatrix} & \text{for } 0 \leq T < 0.4 \\ \begin{pmatrix} -x_2 \\ +x_1 \end{pmatrix} & \text{for } T > 0.4 \end{cases}$$

with T being the dimensionless simulation time and $\boldsymbol{x}$ the position vector with the components x_1 and x_2. In this flow regime, a circular drop of radius $R = 0.3$ and different viscosity is initialized, with its center at $[-1,0]$. So the drop moves clockwise with the fluid, until the flow is fully reversed and the drop moves back to its initial position.

The surrounding fluid should be denoted by Ω_1, the drop by Ω_2. The two fluids have the following viscosities:

$$\mu = \begin{cases} 0.1 & \text{in } \Omega_1 \\ 10 & \text{in } \Omega_2 \end{cases}$$

so the surrounding fluid in Ω_1 is less viscous than the drop. Both fluids are assumed to be massless, meaning $\rho_1 = \rho_2 = 0$. By this, the effects of the inertia and convection are neglected and the flow becomes a Stokes-flow. We choose to do so, because a Stokes flow is known to be fully reversible and after the full period of $T = 0.8$, the bubble shape should be exactly as initialized. By this, we are able to judge the codes abilities of mass conservation and the consistency in calculating the surface tension forces. In standard, free slip boundary conditions are imposed everywhere on the boundary, except on the test case for the effect of the surface tension. Here, no-slip boundary conditions are imposed.

As already described in the overview of the full code in Chapter 3, there are several options which could influence the accuracy of the code regarding the interface shape and especially the mass conservation. To determine the influence of several parameters, we compare the resulting drop shape and the mass conservation to a standard setting. The standard setting is chosen to produce a reasonable but not necessarily very accurate result. The influence of the following parameters will be determined:

- Mesh resolution: The standard mesh with $n_x = 20$ elements per side is refined twice, to $n_x = 28$ and $n_x = 40$.
- Time step size: The main factor for this is the factor θ in the stability condition of the TG3-2S time stepping scheme. Normally set to $\theta = 0.1$, we compare the results of $\theta = 0.2$ and $\theta = 0.05$.
- Reinitialization of the level set: As mentioned, the reinitialization may be necessary for stability reasons. For the Stokes flow, it is deactivated in standard, but we want to show the effect of the reinitialization for every 50 time steps.
- Smoothing of the level set: Another option, which may be used for stability purposes. High frequency oscillations of the level set may be smoothed out. In standard, we turn this off, but the influence on the mass conservation is demonstrated for a relatively large smoothing parameter $\epsilon = 1 \times 10^{-4}$

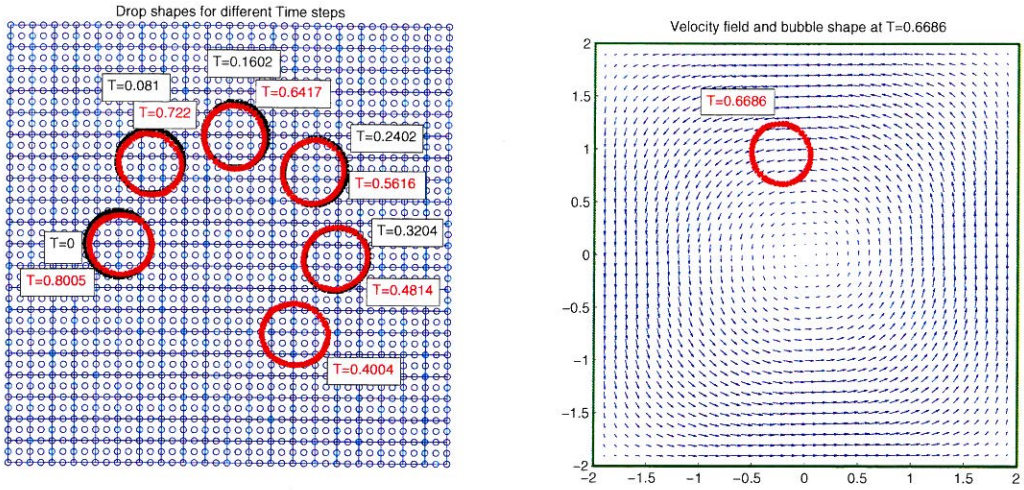


Figure 4.4: Intermediate bubble shapes and initial velocity for drop in circular flow. The plots show the drop shape after every approx. $\Delta T = 0.08$, black for the clockwise direction, red for the counter-clockwise rotation back to the initial position. The second plot shows the velocity and the drop shape for an intermediate state in the backward rotation.

Also, we want to use this example for demonstrating the effect of the surface tension in general. With slightly changed parameters, the drop in the rotating Stokes is calculated with and without surface tension. For both, the mass conservation properties are compared.

In the standard setting, the drop is transported a bit more than half a rotation, before it is directed back. In Figure (4.4) a number of intermediate drop shapes is shown. For first it moves clockwise with (black shapes), before it returns (red shapes).

The results of the standard settings show that the drop does not return completely into its original shape. Due to some lack of accuracy, the shape is not exactly circular anymore. Also, the drop lost mass during the simulation. Figure 4.5 shows the evolution of the area of the droplet over time, which is effectively the mass evolution. Here, in the standard setting, the drop lost approximately 13% of its mass.

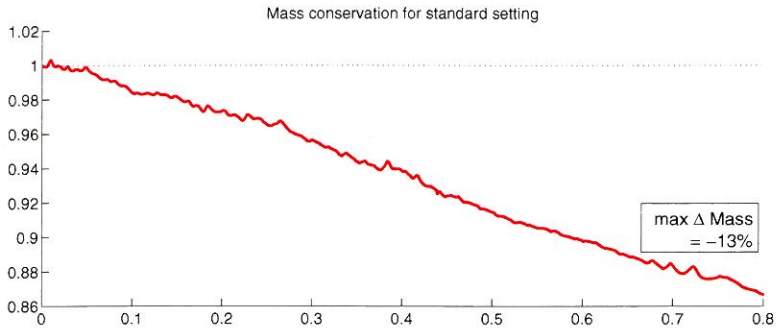
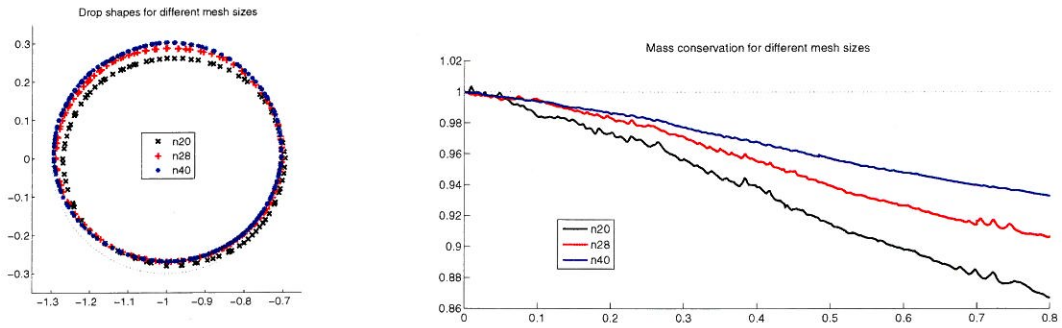


Figure 4.5: Mass conservation for the standard setting for the rotating Stokes flow. The drop lost about 10% of its initial mass.



(a) Shapes (b) Mass conservation

Figure 4.6: Drop shapes and mass conservation for varying mesh resolution. The mass loss reduces from $\Delta\text{Mass} = 13.3\%$ for the coarse mesh (black, $n_x = 20$) to $\Delta\text{Mass} = 9.4\%$ for the $n_x = 28$ mesh (red) and $\Delta\text{Mass} = 6.7\%$ for the finest ($n_x = 40$) mesh (blue)

4.2.1 Mesh resolution

As in other code, the element size of the mesh sets the limits on the accuracy of the results. Here, we calculated the solution for three different quadrilateral meshes ($n_x = 20$, $n_x = 28$ and $n_x = 40$) in order to show the convergence of the solution. In Figure (4.6) the final drop shapes and the mass conservation plot is shown.

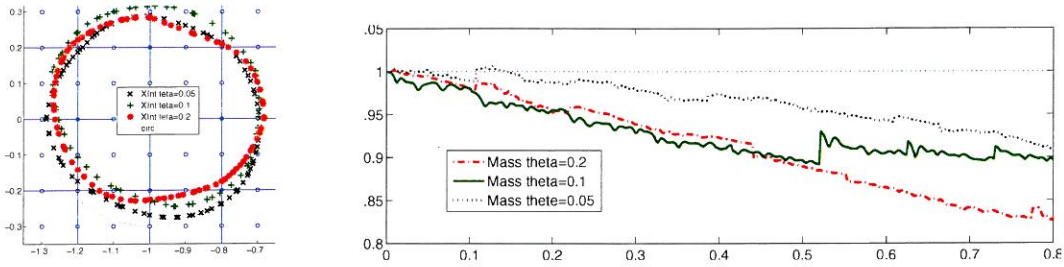
This shows that with mesh refinement, the resulting drop shape after forward and backward rotation becomes closer to the initial shape. Also, the mass conservation

is improved by mesh refinement. The initial mass loss of about 13.3% for the coarse standard mesh ($n_x = 20$) is reduced to 6.7% for the fine mesh with 40 elements per side. So for this first test with just three meshes, we determined a first order mesh convergence ($\mathcal{O}(h^1)$). The reason for this could be the less accurate representation of the surface tension forces, which directly influence the velocity close to the interface and therefore the mass conservation. Also, the full Navier-Stokes system is solved with a first order time stepping scheme, including the inaccuracies brought into the solution by using the n-velocity for the convection term. Further studies of the influencing parameters and other time stepping schemes for the Navier-Stokes equations would give additional information on this.

4.2.2 Variation of the time step size

For any problem, the accuracy of the solution is highly dependent on the time step size. In our code, the time step size is calculated individually for each time step, based on the maximum velocity in the element and the local element size. On the one hand, the time step size has to fulfill the stability condition of the time stepping scheme, which is in our case the TG3-2S scheme. But for higher accuracy and better mass conservation, we reduced the time step size by the factor θ. Choosing $\theta = 1$ means that the stability condition (see Equation (3.6)) is fulfilled exactly. Smaller values of θ reduce the time step size correspondingly.

In Figure 4.7 the effect of the time step size is shown. The bubble shape is more circular for smaller time step sizes. Also, the mass conservation plot shows that for smaller time step size, the overall mass change of is smaller. The mass loss for the large time steps ($\theta = 0.2$) is about 19% of the initial bubble shape. With smaller time steps, this reduces to 13.3% for $\theta = 0.1$ and 9.2% for $\theta = 0.05$. This results in a first order time accuracy ($\mathcal{O}(\Delta t^1)$) for the movement of the level set. Here again, further numerical experiments are needed to consolidate this dependency.



(a) Shapes (b) Mass conservation

Figure 4.7: Drop shapes and mass conservation for varying time step sizes. With smaller time step sizes, the drop shape stays more circular and the overall change of mass becomes smaller

4.2.3 Effect of Reinitialization of the level set

The level-set has to be reinitialized regularly, to prevent the gradient of the level set to become too big, which could cause oscillations on the level set. The reinitialization resets the level set as a signed distance function. But also, for the simple method we implemented here, a change of the position of the interface may be introduced, which corrupts the mass conservation of a calculation. In Figure (4.8) the mass conservation plots of a calculation with and without reinitialization are shown. The big jumps in the drop area are caused by the reinitialization. But nevertheless, for flow conditions, which are strictly oriented into one direction, without reinitialization, the calculation would crash. So the less often the level set needs to be reinitialized, the better is the mass conservation of our code. By the method we choose for calculating the interface curvature, we have weaker restriction on the level set to be a signed distance function, and by this we need to reinitialize it less often.

For the upcoming calculation on the other examples, we choose to reinitialize the level set every 50 time steps. This turned out to be the best trade-off between the stability and mass conservation.

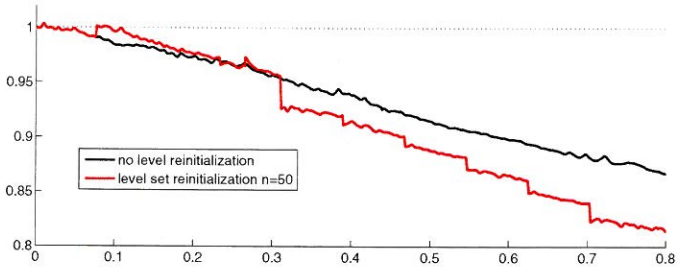
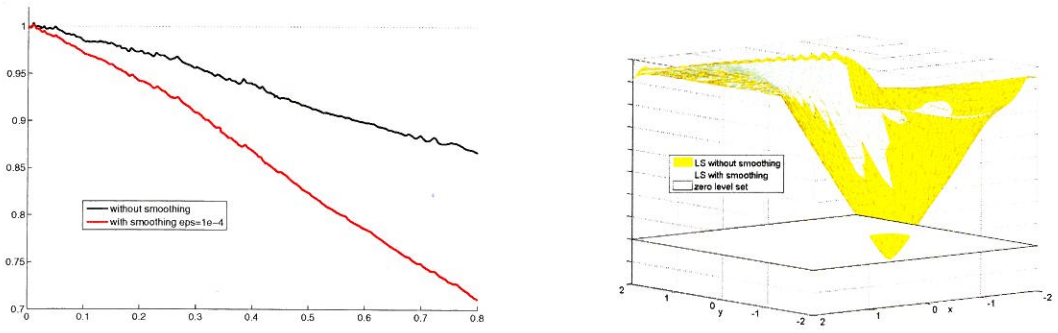


Figure 4.8: Mass conservation for the rotating Stokes with and without level set reinitialization. In this example, the drop mass change with reinitialization is about 1.5 bigger than without.



(a) Mass conservation

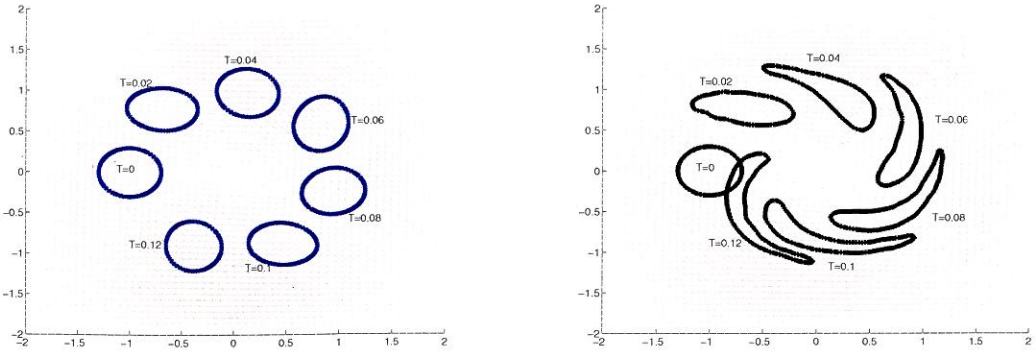
(b) Level sets

Figure 4.9: Mass conservation plot and level set with and without smoothing. The oscillations of the level set near the boundary are smoothed out. In this example, the mass change is about twice as big as without.

4.2.4 Effect of smoothing of the level set

The second part of the maintenance of the level set is the smoothing. To gain a smoother function, we add slight diffusion to the level set function, in order to reduce oscillations of the interface. But this may also smooth out geometrical aspects of the interface and also corrupt the mass conservation properties during the calculation. In Figure (4.9), the smoothing is applied to the current example. As one can see, the level set shape is more regular, but also the drop lost a lot of its initial mass.

In the validation and application of the code to the examples of Rayleigh-Taylor



(a) With surface tension

(b) Without surface tension

Figure 4.10: Effect of surface tension for the rotating Stokes flow: The surface tension keeps the geometry of the droplet more regular.

and bubbles, we normally deactivated the smoothing of the level set, to get better mass conservation properties.

4.2.5 Effect of surface tension

To generally show the effect of the surface tension, the set-up has been slightly altered. Instead of free slip boundary conditions, now no slip is imposed everywhere on the boundary. By this, the fluid is undergoing a higher shear stress, which will increase the deformation of the drop itself.

Also, we decreased the fluids viscosity and calculate for a longer rotation period. By this, the drop should really be sheared of its original circular shape, if no surface tension is acting. In contrast, using a relatively large surface tension coefficient, the drop stays in a nearly circular shape. In Figure (4.10) the drop shapes are shown for several time steps.

In the mass conservation plots for this case (Figure (4.11)), one can see that the mass conservation properties are better for the flow with surface tension. The reason

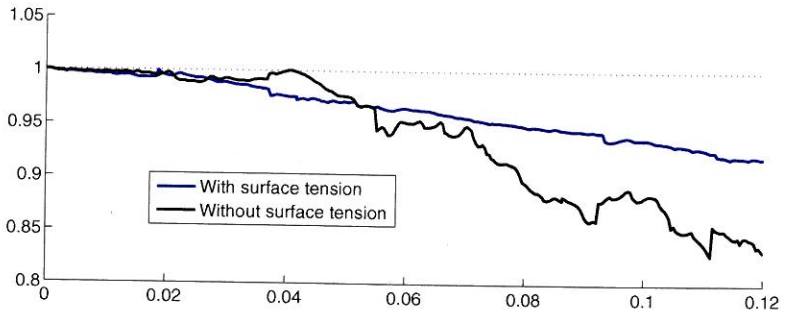


Figure 4.11: Effect of surface tension on the mass conservation for the rotating Stokes flow: Due to the more regular shape, the interface length of the case with surface tension is smaller and mass is better conserved.

for this is, that although both of the drop start with the same area, by moving out of the circular shape, the case without surface tension gets a much longer interface. This is due to errors caused by moving the level set effect on the full interface length. For longer interfaces, the mass conservation properties are harder to hold.

Due to the high mass loss in the calculation without surface tension, reversibility of the Stokes flow does not count anymore. Therefore no results for the reverse flow are shown here.

4.3 Rayleigh-Taylor instability

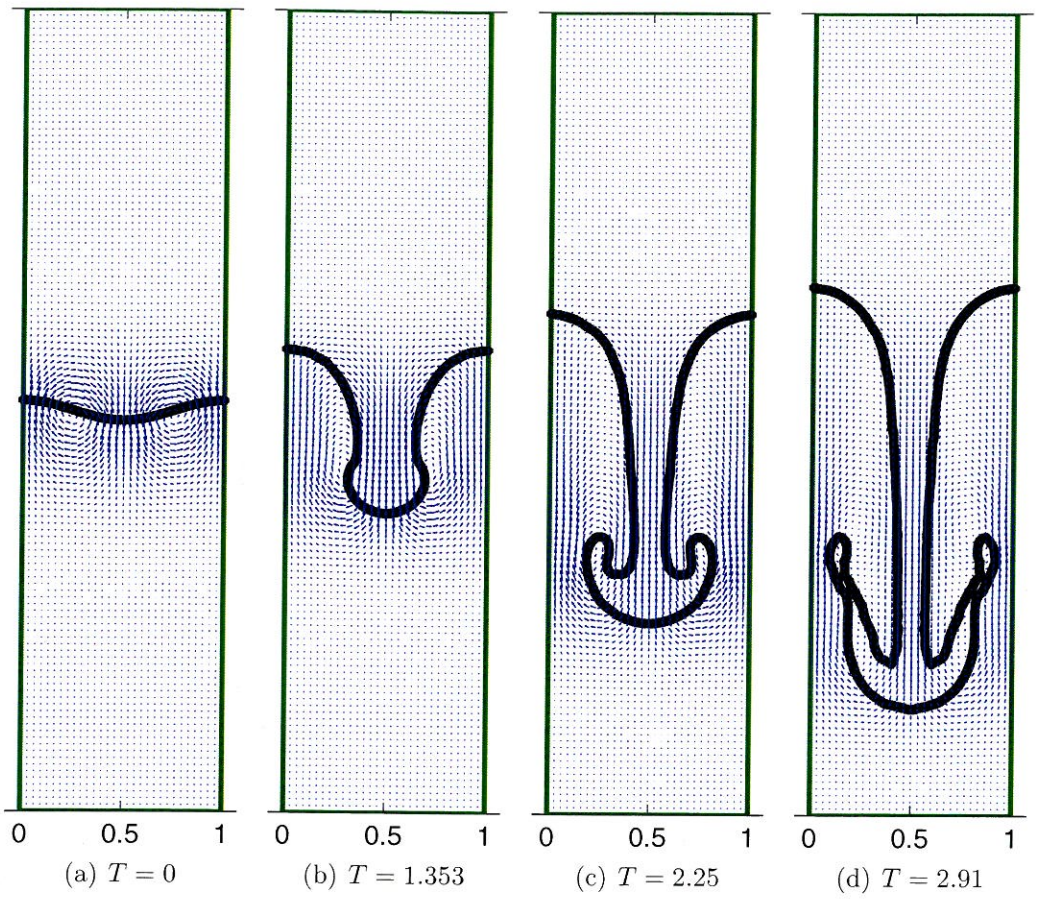
The Rayleigh-Taylor instability is a standard example for immiscible two-phase flows and a large number of numerical experiments have been performed on this problem. (e.g., (Marchandise et al., 2007; Lock et al., 1998)). Here, we follow the settings which have also been used and discussed in (Smolianski, 2001). Since the two acting fluids are physically relatively similar, this setting is not very problematic in terms of stability. So the results presented in this section, have been produced without using the stabilization and smoothing techniques. The re-initialization of the level-set was applied every 50 time-steps.

The setting is in a rectangular domain of size $[0,1] \times [0,4]$. Initially the interphase is described by the equation

$$y_\Gamma(x) = 2 + 0.05\cos(2\pi x)$$

and the initial velocity field is zero everywhere. The boundary conditions are no slip on the top and bottom of the domain and free slip on the vertical walls. As physical parameters, we assume both fluids to be of the same viscosity of $\mu_1 = \mu_2 = 3 \cdot 10^{-3} kg/(m \cdot s)$ and the fluid densities are set to $\rho_1 = 1.2 kg/m^3$ and $\rho_2 = 0.17 kg/m^3$. As in (Smolianski, 2001) we assume the gravitational acceleration to be of size $g = 1 m/s^2$. For this calculation we use a relatively coarse mesh of 15×60 elements of type $Q2Q1$.

For the first simulation, we set the surface tension coefficient to zero. Here, in the beginning the heavy fluid sinks in a droplet-shape into the less dense fluid. In this phase, one could imagine the interphase shape to be part of a periodic interphase of two fluids penetrating. Then the drop begins to form the characteristic mushroom shape on the front and starts to fade out in the sides. This fade-out can be traced back to the Kevin-Helmholtz instability.



(a) $T = 0$ (b) $T = 1.353$ (c) $T = 2.25$ (d) $T = 2.91$

Figure 4.12: Rayleigh-Taylor instability without surface tension shows how the interphase evolves to a mushroom shape with Kevin-Helmholtz instability on the sides

In comparison of the shapes of the interphase we gain in our calculation (see Figure (4.12), we can state a general agreement with the results given by (Smolianski, 2001) and (Puckett et al., 1997), but there are also differences. For example, the mushroom shape of our results is earlier faded out at the sides, whereas it stays more regular in the calculation of Smolianski, which are displayed in Figure (4.13). Nevertheless, the mass conservation is of acceptable order, as Figure (4.14) shows. The maximum change of volume is about 0.25% of the full domain.

For the mass conservation, the effect of the reinitialization of the level-set is dominant. As already mentioned, the level set was re-set to a signed distance function

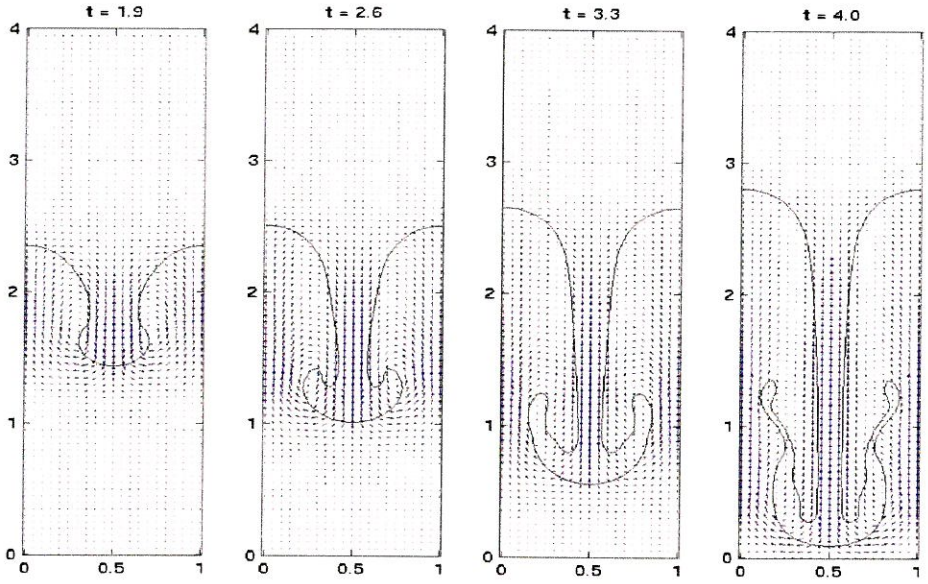


Figure 4.13: Results for Rayleigh-Taylor-instability published in (Smolianski, 2001)

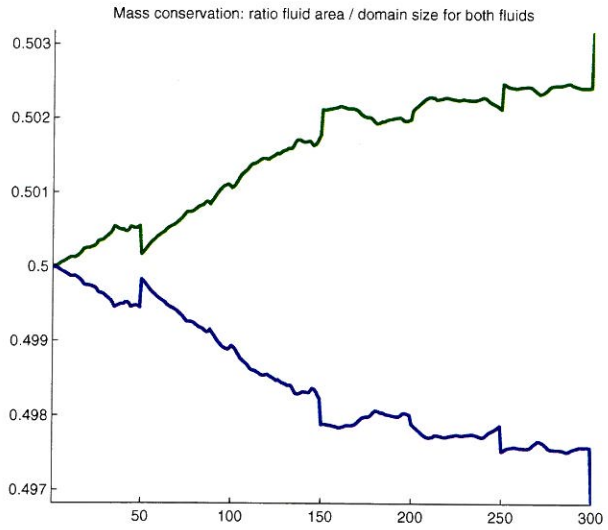
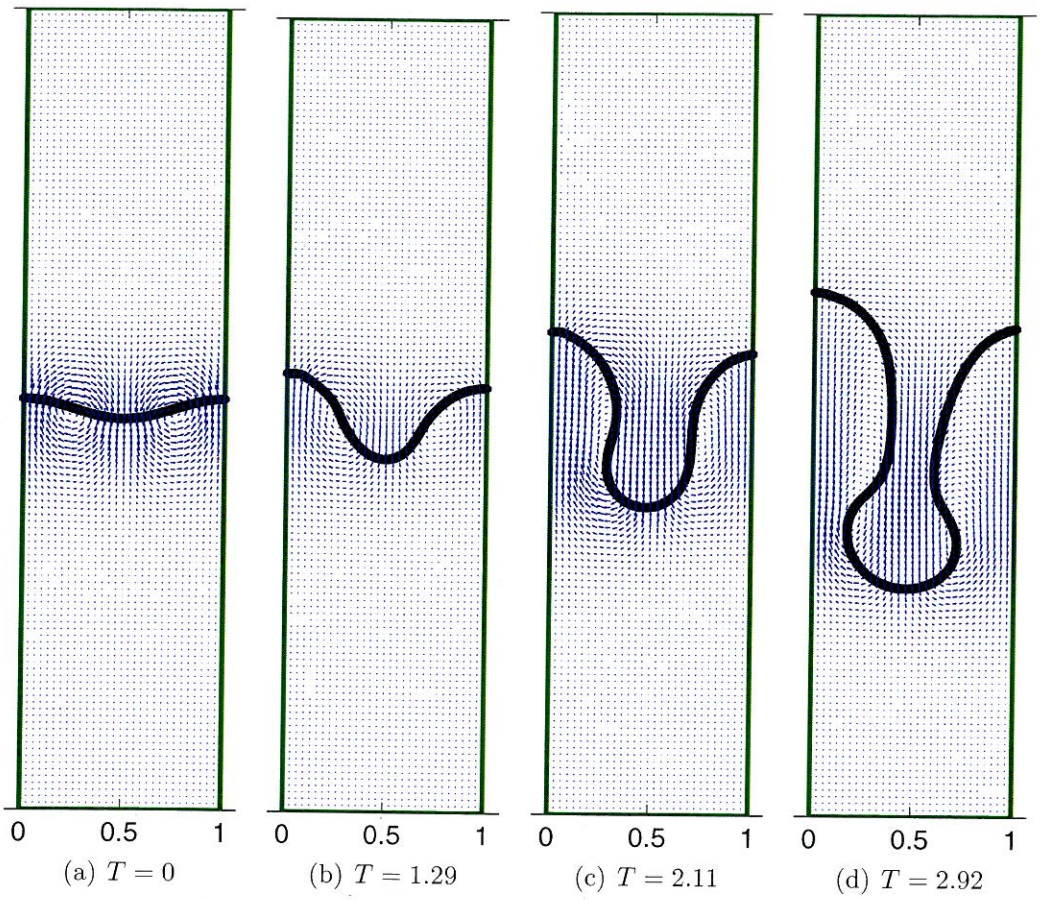


Figure 4.14: Mass conservation for the Rayleigh-Taylor-instability without surface tension. The maximum mass change is about 0.25% of the full domain

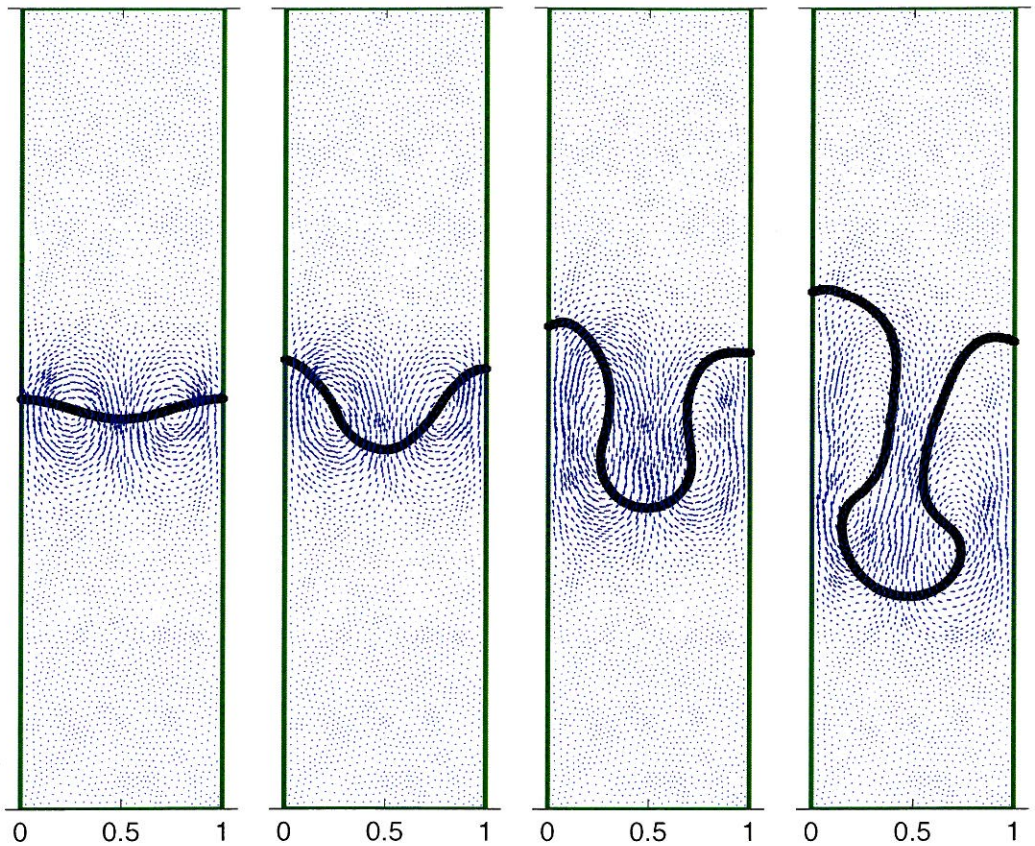


(a) $T = 0$ (b) $T = 1.29$ (c) $T = 2.11$ (d) $T = 2.92$

Figure 4.15: Rayleigh-Taylor instability with surface tension coefficient $\tau = 0.015$ behaves slower and more regular. But due to inaccuracies in the surface-tension term, it becomes non-symmetric.

every 50 time-steps. This explains the "steps" in the mass conservation plot on the time-step 50, 100, 150 and so on. But the reinitialization is needed in this case, because as the denser fluid drops down, the gradient of the level set function becomes very steep. It was also tested to reinitialize the level-set less often, e.g., every 75 or 100 time-steps. But for these settings, high oscillations of the level set appeared, which have been strong enough to cross the zero-level. These had been interpreted as new, artificial interphases and corrupted the solution.

Secondly, we add slight surface tension to this problem. In (Smolianski, 2001) a way



(a) $T = 0$ (b) $T = 1.26$ (c) $T = 2.11$ (d) $T = 2.94$

Figure 4.16: Rayleigh-Taylor instability with surface tension coefficient $\tau = 0.015$ on an unstructured mesh. Here, also the symmetry of the case is corrupted.

to calculate the critical value for the surface tension coefficient is mentioned and for this setting stated to be around $\tau_{\text{crit}} \approx 2.6 \cdot 10^{-2}$. This would be the maximum surface tension coefficient for the instability to evolve. Then, we choose value of $\tau = 0.015$, and calculate the instability with the same coarse mesh of 15×60 elements.

The results of the Rayleigh-Taylor instability with surface tension can be seen in Figure (4.15). Generally, since the surface tension hinders the heavy drop to penetrate the light fluid, the flow is slower than the flow without surface tension. Also, due to the surface tension, the shape is more regular and smoother, but also non-symmetric.

Nearly the same results are gained, if the Rayleigh-Taylor-instability is calculated

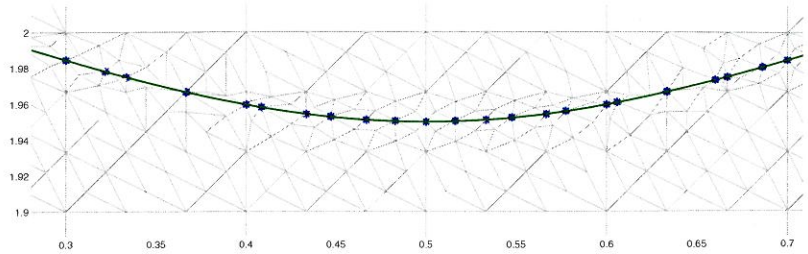
on a unstructured mesh. The results for a triangular unstructured mesh with approximately the same number of degrees of freedom like the structured used before can be seen in figure Figure (4.16). Here, again, the solution becomes non-symmetric by the representation of the interphase on the reference mesh.

So, in both meshes, structured and unstructured, the surface tension force has a non-symmetric influence on the solution. This effect can not be seen in the same calculations, which are performed by Smolianski. There, the shape is also more regular, but it stays symmetric. This means, in the results of Smolianski, the surface tension does not introduce non-symmetry in the solution.

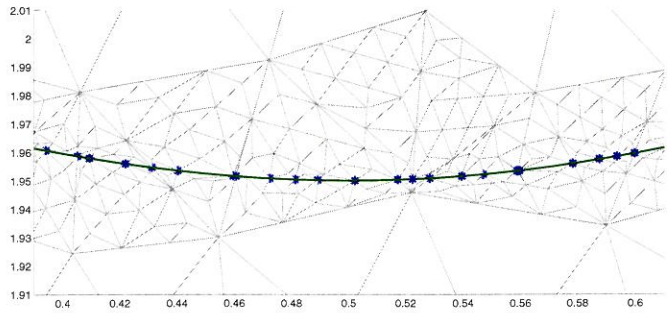
In our code, the non-symmetry is introduced by the representation of the interphase with the help of the zero-level-set points of the integration cells. Inside each of the mesh elements, the reference cells are initialized non-symmetric, therefore the points, which will be used for the interface approximation, are also not symmetric. This causes slightly non-symmetric polynomials of the interface approximation, and by this the surface tension forces are not symmetric anymore. In Figure (4.17) the points which are used for the polynomial approximation are shown for the center of the cos-shape of the initial interphase. As one can see, for each of the sides of the domain, the points representing the interphase differ.

To demonstrate the mass conservation for the different the unstructured and the structured mesh, Figure (4.18) plots the area for the denser fluid for the first three seconds. In general it can be said that the mass conservation is still acceptable for this size of the mesh elements. In maximum, over the first three seconds, the mass changes about 0.43% of the full domain size for the unstructured mesh and about 0.25% for the structured meshes.

The structured mesh shows a lower change of mass than the unstructured. The reason for this is the time step size for each individual time step. This is driven by the maximum ratio of the velocity in one element by a characteristic element size. For the unstructured mesh, this characteristic length is bigger than for the structured mesh.



(a) Structured Mesh



(b) Unstructured Mesh

Figure 4.17: Representation of center of the interphase for the Rayleigh-Taylor instability on a structured and an unstructured mesh. The reference mesh causes a non-symmetric distribution of the interphase points, which introduces non-symmetry in the solution

So each time-step has been bigger, and therefore the error caused by the movement of the level set is bigger. This explains the higher mass change for the unstructured mesh.

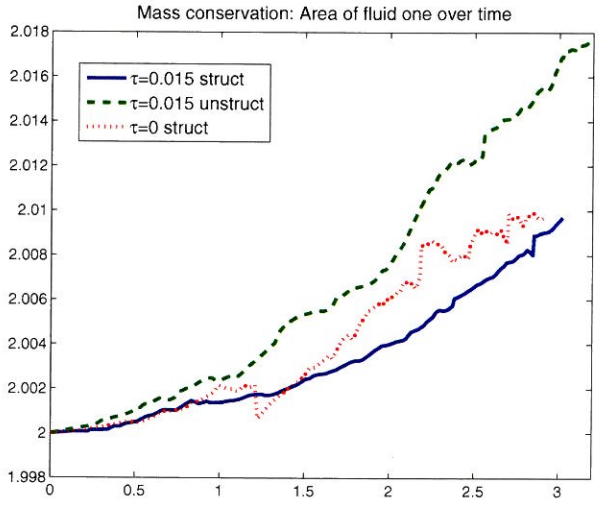


Figure 4.18: Mass conservation for the Rayleigh-Taylor-instability with and without surface tension, on structured and unstructured meshes. The untructured mesh has the higher change of mass. Maximum mass fluctuation approximately 0.43% of the full domain for the unstructed mesh. The structured mesh shows only slight differences between the cases.

4.4 Rising bubbles

The rising bubbles example is another standard test case for immiscible two-phase flows with surface tension. It has the advantage that the light-weight bubble rising in a more dense fluid evolves to a final shape and the rising speed converges to a certain value. In this final state, the rising buoyancy forces balance the viscous drag of the bubble. Also, since the conditions can be reproduced in experiments, this is one of the few numerical examples whose results can be compared to experimental data.

Herein, we compare our numerical results with those given the paper by (Hua and Lou, 2007), who compared their numerical results with experimental data published in the early 1980s (Bhaga and Weber, 1981).

Basically, the final bubble shape depends on two non-dimensional characteristic numbers, which describe the flow regime, the Reynolds number and the Bond number, the latter is also known as Eotvos number. In (Hua and Lou, 2007) a definition for these characteristics, which only depends on the parameters of the surrounding liquid – denoted by the subscript L is used:

$$Re = \frac{\rho_{\mathrm{L}} g^{(1/2)} D^{2/3}}{\mu_{\mathrm{L}}} \text{ and } Bo = \frac{\rho_{\mathrm{L}} g D^2}{\tau}$$

Here, g represents the gravitational acceleration, D the initial bubble diameter and the parameters of the surrounding liquid are μ_{L} and ρ_{L}. The influence of the surface tension is introduced by the surface tension coefficient τ in the Bond-number. In general, it can be said that for higher Reynolds numbers, the final bubble shape is less regular. Secondly, the smaller the Bond number is, the higher is the influence of the surface tension and for higher Bond numbers, the bubble shape becomes less regular.

The calculations made in this section are all done on a domain of size $[0, 1] \times [0, 3]$ and the boundary conditions are set to free slip. The viscosity ratios of liquid/bubble

are set to $\mu_1/\mu_2 = 100$, and the density ratios are set to $\rho_1/\rho_2 = 1000$. Initially the bubble is positioned at the point $p = [0.5; 0.5]$ and the initial velocity is assumed to be zero everywhere. To avoid the influence of the top wall, the calculation stops latest, when the top of the bubble reaches $y = 2.75$.

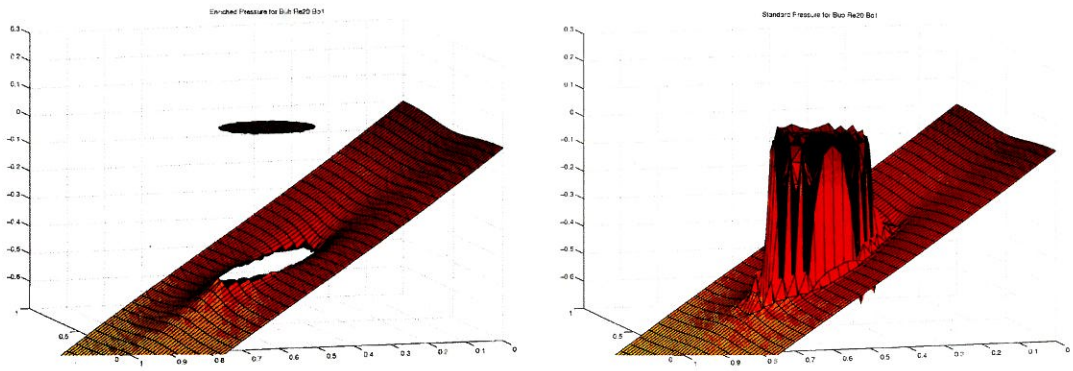
4.4.1 Rising bubble with $Re = 20$ and $Bo = 1$

To determine the improvements in the results by enriching the velocity and the pressure, we calculate a bubble rising in a setting, where the bubble shape is nearly circular, due to a very high surface tension. For this, the parameters are chosen such that the Reynolds number becomes $Re = 20$ and the Bond number $Bo = 1$.

In Figure (4.19), the pressure after the first time step is shown with and without enrichment. As already seen for the non-physical example (see Figure (4.2)), the pressure is represented much better due to the enrichment. With enrichment, nearly no oscillations appear close to the interphase and the pressure jump is clearly resolved. Without enrichment, in contrast, the pressure is continuous and hardly able to represent the pressure jump. By this, pressure oscillations appear near the interface.

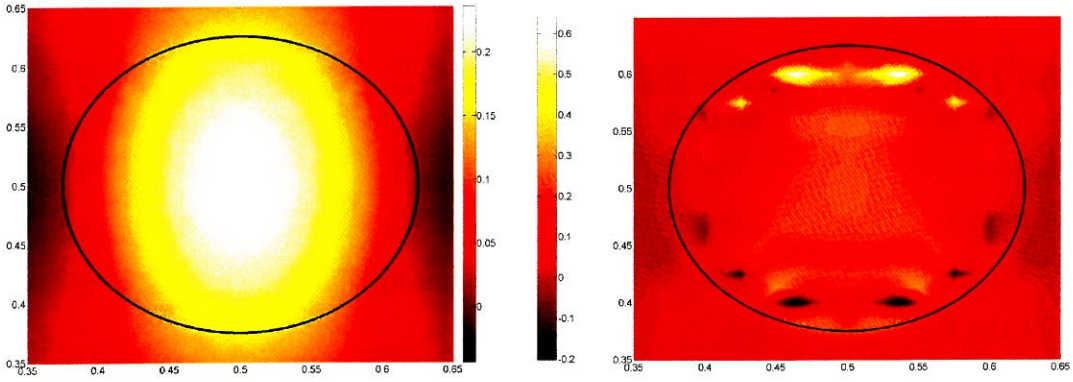
Also, the velocity enrichment has a big improving effect on the solution. In Figure (4.20), a plot of the y-velocity in the bubble is shown as an interpolated contour plot, with and without enrichment. In general, the fluid y-velocity is the highest in the center of the bubble. This is represented clearly by the bright shade of the enriched calculation. In contrast to this, for the not-enriched velocity, high oscillations appear in the vicinity of the interface. This is caused by the insufficient representation of the non-smooth velocity at the interface.

In the calculation without enrichment, the oscillations of the velocity corrupt the solution and a final state can not be calculated with our code. In contrast, for the same mesh and with enrichment, a proper solution for the bubble rising in the fluid



(a) With enrichment (b) Without enrichment

Figure 4.19: Pressure of the first time step for the bubble with $Re = 20$ and $Bo = 1$, with and without enrichment. Due to the enrichment, the pressure jump is represented much better and nearly no oscillations close to the interface appear.

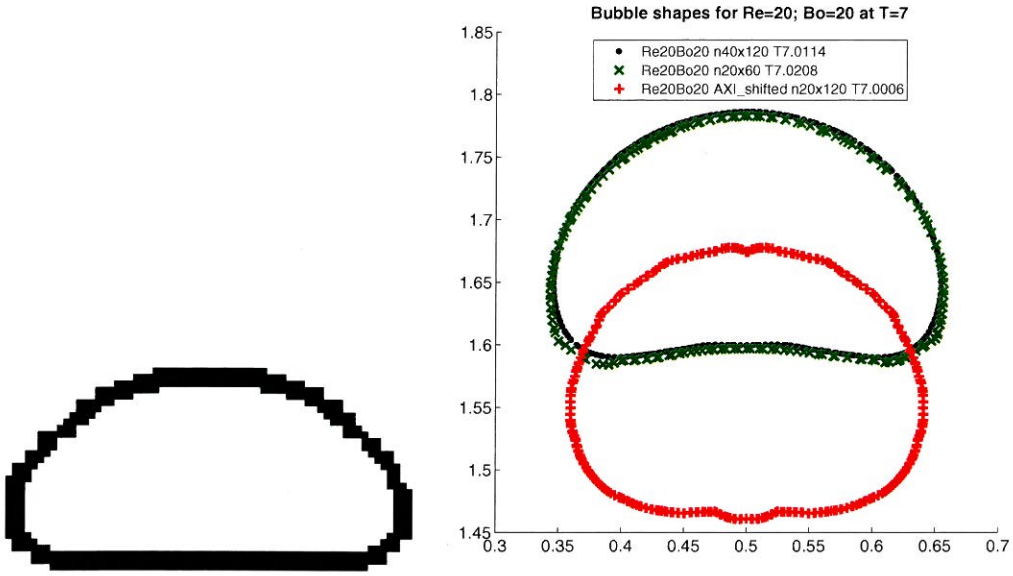


(a) With enrichment (b) Without enrichment

Figure 4.20: y-Velocity contour plot for first time step of the bubble with $Re = 20$ and $Bo = 1$. The non-smooth velocity is much better represented with enrichment. Without, high oscillations appear near the interface.

is found. This shows the abilities of the enrichment to improve also the stability of such calculations with strong surface tension effects.

For the calculation with enrichment, the final bubble shape becomes a nearly exact circle. This is due to the very high surface tension. For other settings, the final bubble



(a) given shape of (Hua and Lou, 2007)

(b) Results of method at hand for 2D (black, green) and axis-symmetric formulation

Figure 4.21: Bubble shapes for $Re = 20$ and $Bo = 20$ at Time=7. On the left the predicted by Hua and Lou, right the calculated with the method at hand. Nearly no difference between coarse and fine mesh, but axissymmetric calculation differs strong

shape can show, how accurate the interaction between the surface tension forces and each of the fluids is modeled.

4.4.2 Rising bubble with $Re = 20$ and $Bo = 20$

For a comparison of our method with the method of (Hua and Lou, 2007), we start with a bubble setting of $Re = 20$ and $Bo = 20$. Following the shape list given in there, the bubble should finally reach a semi-circular-like state, with a flat horizontal bottom and a nearly circular head. Figure (4.21(a)) shows the predicted bubble shape in the paper.

In Figure (4.21(b)) the resulting bubble shapes of our code are shown. The solution is calculated for the two possibilities of an standard two-dimensional case and for

axis-symmetric setting. For the standard discretization, a coarse mesh with $[20 \times 60]$ elements and a finer mesh with $[40 \times 120]$ elements have been used. The difference between the fine and the coarse mesh is marginal, such that we can assume the mesh resolution of the fine mesh to be appropriate. The elements of the axis-symmetric mesh are of the same size as in the fine case.

What becomes obvious is that the axis-symmetric case not only rises slower, but also the final bubble shape is more circular, as if the curvature was over-predicted. Also, small kinks in the location of the rotational axis (for plotting shifted to $x = 0.5$) show that the symmetry condition is not completely fulfilled for this formulation. One reason for this appears to be, that the least squares approximation of the interface is not able to fully resolve the sharp edge on of the interface close to the axis of rotation. Further studies on the behavior of the code for the axis-symmetric settings are necessary.

Comparing the final bubble shape of the 2D-formulation with the shape presented in the paper, it can be said that they match relatively well. The curvature on the bottom side of the bubble is very small and also the upper side of the bubble has a close-to circular shape. This matches to the predicted bubble shape of the paper. So for this case, the method proposed gives qualitatively good results.

In Figure (4.22) a mass conservation plot of the bubble is given for the different meshes of the two-dimensional case and the axis-symmetric setting. In general, the reinitialization steps of the level set becomes obvious as big steps in the bubble area. This shows again, that the reinitialization changes the interface position and therefore the bubble mass. Although it was shown that the shapes of the bubble do not differ too strong between the finer and the coarser mesh, here we see that the fine mesh produces a much better mass conservation results. Compared to its starting size, it gains around 1.5% of area for the coarse mesh, whereas it only looses 0.649% of its mass in the finer mesh.

Moreover, for this setting with less surface tension than before, we also performed a calculation without enrichment, to show the influence of the enrichment even on

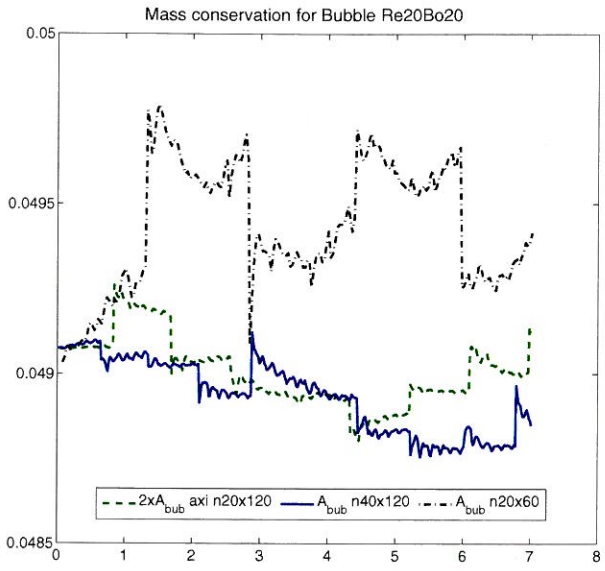
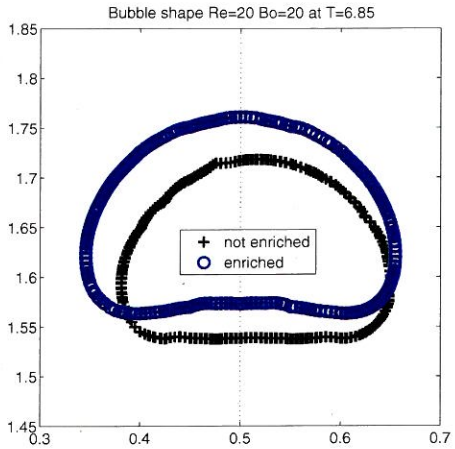
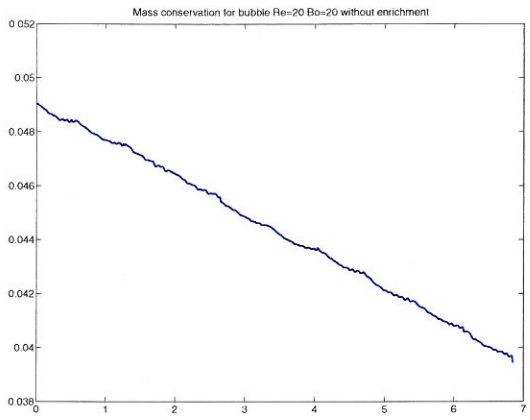


Figure 4.22: Mass conservation of bubble with $Re = 20$ and $Bo = 20$. The maximum change of area is about 0.649% of bubble area for $n = 40 \times 120$, but also the change of the interface due to the reinitialization of the level-set every 50 time-steps can be identified clearly

cases with less influence of the surface tension. As shown in Figure (4.23(a)), the final bubble shape after the rising time of $T = 6.85$ slightly differs from the one calculated with enrichment. The interface representation has more oscillations and the full bubble moved out of the center, slightly to the right. The mass conservation also shows a lack of accuracy for the calculation without enrichment. In Figure (4.23(b)), the mass conservation plot shows that over time the bubble lost about 20% of its initial area. To get the same accuracy as for the calculation with enrichment, the mesh would have to be strongly refined. This emphasizes again the importance and the abilities of the enrichment for the cases of two-phase flows. The mass loss also results in a smaller rising velocity, that is why the bubble without is under the bubble with enrichment.



(a) Bubble shape $T = 6.8$



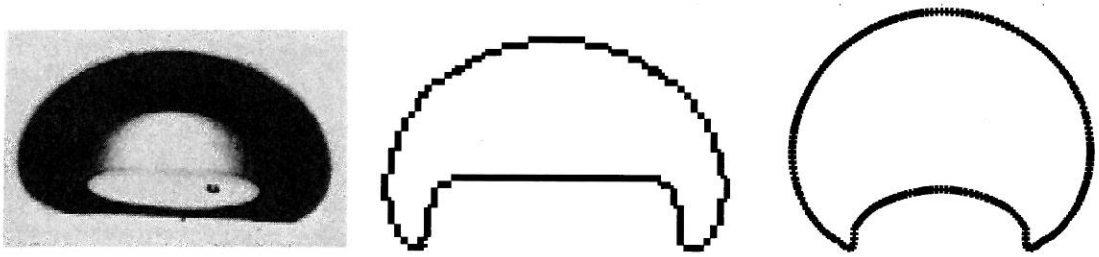
(b) Mass conservation

Figure 4.23: Bubble shape and mass conservation for $Re = 20$ and $Bo = 20$ without enrichment: The interface shape has more oscillations and also the interface moved out of symmetry. The mass conservation is much worse without enrichment; the bubble lost around 20% if its initial area

4.4.3 Rising bubble with $Re = 15$ and $Bo = 243$

Also, we applied our method to the setting of $Re = 15$ and $Bo = 243$. The high Bond number marks setting to be less dominated by surface tension. In Figure (4.24) the final bubble shape of the method proposed here is compared with the result of the experiments of (Bhaga and Weber, 1981), but also with the predicted shape of (Hua and Lou, 2007).

For this case, the the bubble shape calculated with our code does not match the shape of the experiments, neither the prediction of (Hua and Lou, 2007). Our result is more circular on the upper side of the bubble, and we have strong kinks in the lower side. In general, the expected result should be more flat. It seems as if the surface tension effect is on the one hand over-predicted in the upper area, whereas it is under-predicted where the kinks appear. This effect is not caused by the fact that the bubble did not reach its final state yet. An overlay of the bubble shape of a



(a) Final bubble shapes of experiments of (Bhaga and Weber, 1981)

(b) Result of (Hua and Lou, 2007)

(c) Result of the present code

Figure 4.24: Final bubble shapes for $Re = 15$ and $Bo = 243$ of an experiment, the method of (Hua and Lou, 2007) and the method at hand. Our calculation shows a more circular bubble shape, and a stronger kink on the bottom

earlier state has shown exactly the same bubble shape. Also, in another calculation on a finer mesh, no big difference on the bubble shape have been observed. So the inaccuracy for calculating the effect of the surface tension seems to be a principal error of our method, which can not be improved by further mesh refinement.

Anyways, the mass conservation plot for this test case in Figure (4.25) also shows that we get acceptable mass conservation properties in our code. The maximum change of area is about 0.6% of the initial bubble area.

4.4.4 Stability limitations for higher Reynolds numbers

The examples discussed above are characterized by relatively small Reynolds numbers. By this, the Buoyancy force on the bubble is relatively small compared to the viscosity and the resulting velocity is also relatively small. During test runs, it was experienced that with the method of moving the level set we have now, for higher Reynolds Numbers, problems with the stability appear. For less viscous flows, spurious currents appear close to the interface, which could even become the main driving velocity of the set-up. This results in corrupted movement of the level set, with oscillations. If those oscillations pass the zero level of the level set, new and non-physical interfaces

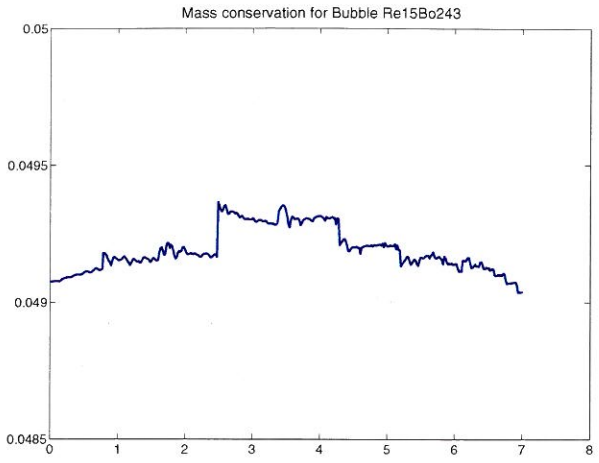


Figure 4.25: Mass conservation of bubble with $Re = 15$ and $Bo = 243$ with a 0×120-mesh: Good mass conservation properties with a maximum area being 0.59% higher than the initial bubble shape

Density	$\rho_1 =$	1000
	$\rho_2 =$	1
Viscosity	$\mu_1 =$	$1 \cdot 10^{-3}$
	$\mu_2 =$	$1 \cdot 10^{-5}$
Surface tension	$\sigma =$	$7.28 \cdot 10^{-2}$
Gravity	$\mathbf{g} =$	9.81

Table 4.2: Initial parameters for water-air example

are introduced. These interfaces have large curvatures and by this, high velocities appear nearby and further corrupt the solution.

In this section, we want to show the limitations of the code for flows with very low viscosity. Starting from the setting of pure water and air, we demonstrate how the solution gets corrupted. Afterwards, we increase the viscosity and reduce the density, until the solution is stable and the oscillations of the level set do not corrupt the solution within the first 50 time steps. The physical parameters of this setting can be found in Table 4.2. Following the definition of the Reynolds number and Bond number in (Hua and Lou, 2007), this would mean a flow with $Re = 1.24 \cdot 10^6$ and $Bo = 3.37 \cdot 10^4$. This flow is very sensitive on spurious currents, since small

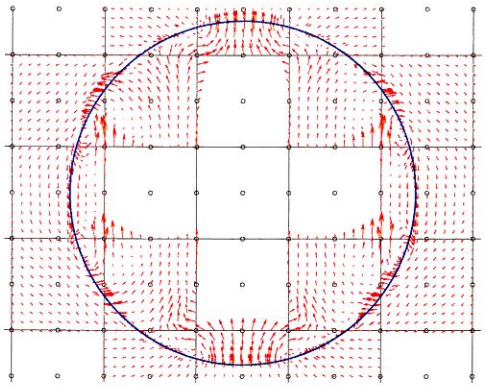
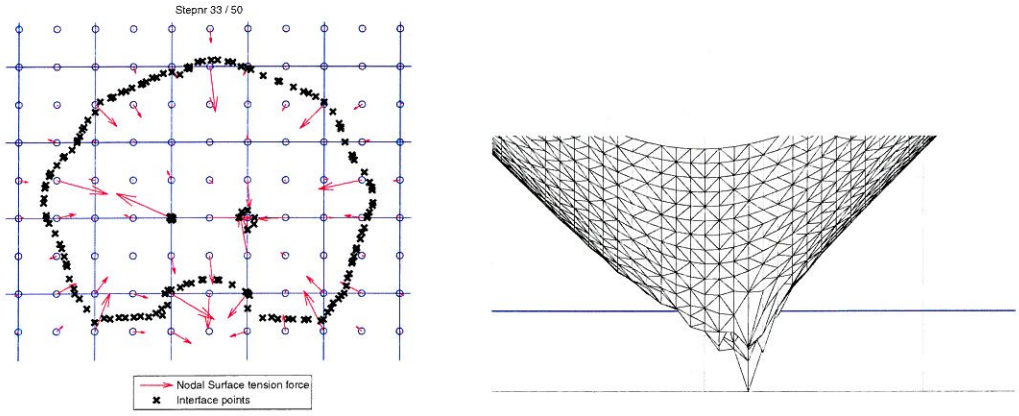


Figure 4.26: Spurious velocity for water/air flow, without surface tension. The discontinuity causes non-physical velocities close to the interface

inaccuracies cause high changes of the velocity. In Figure (4.26) the velocity after the first time step indicates the problematic flow situation. Note that the non-physical velocity is introduced only be the different material properties at the interface, since this figure was produced without influence surface tension. Inaccuracies due to the surface tension would add to this. For this setting, the interface was corrupted within a few time steps.

We increase the viscosity, to reduce the Reynolds number, until a proper solution can be found and the solution does not get corrupted anymore. Such a corrupted interface is shown in Figure (4.27(a)), and the corresponding corrupted level set in Figure (4.27(b)). This is the result for $Re = 1.24 \cdot 10^4$, but problems like this also appear for smaller Reynolds numbers.

Finally, with changed settings with much smaller Reynolds numbers, we were able to get a solution. But for a general statement of the stability limit, we had been unable to manifest a critical Reynolds number, which could be used as a stability limit. The oscillations are dependent on the mesh resolution, as well as on the time step size. Smoothing the level set and re-initialization, before the oscillations become too large are necessary to keep the stability. But nevertheless, those corrupting oscillations of the level set are the limiting factor for the application of this code.



(a) Interface and nodal surface tension

(b) Level set with oscillations

Figure 4.27: Corrupted interface and level set for rising bubble with $Re = 1.24 \cdot 10^4$ after a few time steps. Oscillations of the level set cause smaller interface.

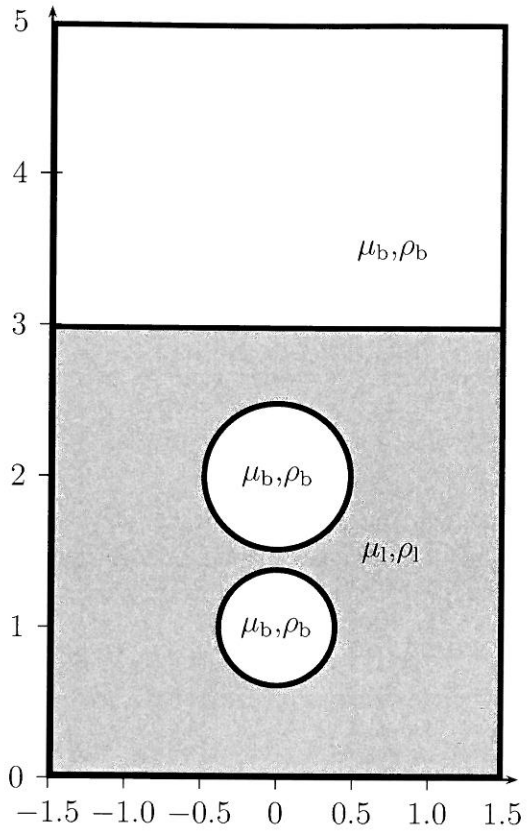


Figure 4.28: Initial setting of two bubbles rising to a free surface, with the physical parameters μ_b, ρ_b for the bubble fluid and μ_l, ρ_l for the liquid fluid.

4.5 Joining rising bubbles with free surface

In the field of rising bubbles, another popular test case is the joining of two bubbles or a bubble joining a free interface or the combination of both. This example gives a good insight, on how good a model can handle big changes of the topology of the interface, especially in terms of mass conservation. Results for this test case are, for example, presented in (Tornberg and Enquist, 2000), (Marchandise et al., 2007), (Chessa and Belytschko, 2003) and (Smolianski, 2001).

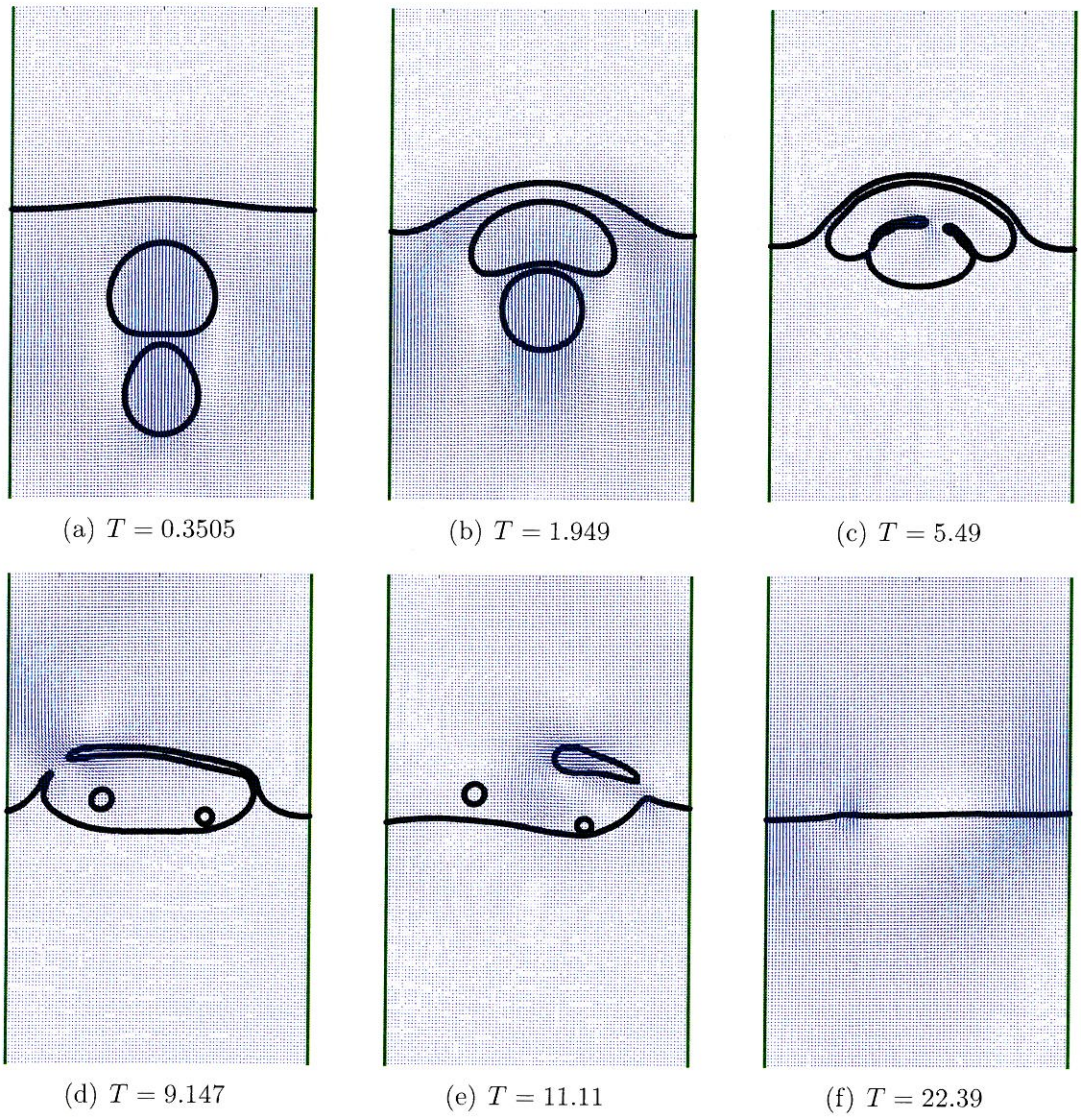
Here, we follow the settings of (Tornberg and Enquist, 2000) of two bubbles joining and then rising to a free surface. The initial setting is illustrated in Figure (4.28). The domain is of size $[-1.5, 1.5] \times [0, 5]$ and the free surface is defined at $y = 3$. The smaller

bubble is situated at $[0, 1]$ and its radius is 0.4, and the bigger bubble is positioned at $[0, 2]$ with a radius of 0.5. As initial condition, both fluids are at rest and on the boundaries, free-slip conditions are imposed. For this test case, a structured mesh with 45×75 elements of type $Q2Q1$ is used; this results in a mesh of 13741 nodes. The physical parameters are set such that the ratio of the densities ρ_l/ρ_b is about 100 and the ratio of the viscosities μ_l/μ_b is 2. The surface tension coefficient is set to $\tau = 0.1$ and the gravitation to $g = 1$. In the calculation, the level set is reinitialized every 50 time steps and the smoothing of the level set is deactivated, as well as the stabilization of the solution. An upper limitation of the interface curvature, as it was used in (Tornberg and Enquist, 2000), was not necessary for our code.

The principal physical behavior of the two bubbles merging to a free surface can be summarized like following: Since the smaller, lower bubble is in the wake of the bigger, it rises faster. By this, the distance between both gets smaller, until it is zero, and the separating layer of the liquid between them gets thinner, until it breaks up. In this moment, the two bubbles join. The same happens, when the union of both bubbles approaches the free surface: The separating layer of liquid becomes thinner, until it breaks up. By this, a sharp edge of the interface appears on the tips, which by surface tension highly accelerates the fluid. Once these kinks are flattened out, the solution reaches the final state, where a single interface separates the fluid of the bubble with the liquid fluid and both fluids come to rest. The sequence in Figure (4.29) shows the results of our code on this.

Qualitatively, our solution shows the effects of the two bubbles joining with a free surface in a reasonable manner. As already experienced for the Rayleigh-Taylor instability, there are effects in the surface tension, which lets the solution become non-symmetric.

Secondly, in our code, the merging of the two bubbles and the merging with the free surface takes place later. The separating layer stays longer between the two areas of the same fluid and prevents them from merging. As an effect, both bubbles accumulate



(a) $T = 0.3505$ (b) $T = 1.949$ (c) $T = 5.49$

(d) $T = 9.147$ (e) $T = 11.11$ (f) $T = 22.39$

Figure 4.29: Resulting shapes of two bubbles rising to a free surface. The result is non-symmetric, but reasonable. The separating layer between the bubbles or the bubble and the surface stays very long

first directly under the free surface and after that, they merge to one bigger bubble, before the free surface breaks. Moreover, the separating layer does not only break once in the center and is then pulled back. Instead, it breaks at two different places and forms then another bubble for its own. The two additional bubbles in the time $T = 9.147$ are formed out of the separating layer between the two bubbles, and then,

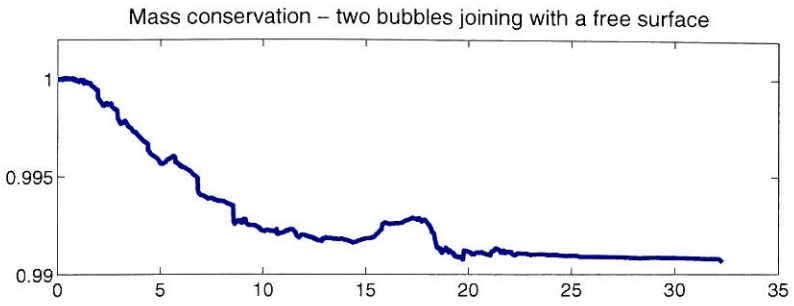


Figure 4.30: Mass conservation plot of two bubbles merging with a free surface: The bubble fluid only loose 1% of its initial area, although the topology massively changed.

by surface tension formed a nearly circular drop. Later, these drops fall to the free surface and join with the fluid below. Finally, the fluids form one single interface and come to rest.

To have a look on the mass conservation, in Figure (4.30) the ratio of the area of the bubble liquid to its initial area is plotted. By this one can see, that although there is a huge change in the topology of the interface, the mass is conserved properly. In maximum, the bubble fluid has lost less than 1% of its initial mass.

This example again shows the codes' abilities of generally resolve the effects of two phase flows with surface tension properly. For a huge variety of examples, we gained qualitatively reasonable results, with good mass conservation properties. But we have also shown limitations in terms of accuracy and stability, which may cause problems in the application to more complex problems. On these issues, further studies are required. Also, in axis symmetric cases, the symmetry condition at the rotational axis cause problems. But nevertheless, in the current state, the code is already able for a close-to-industry application. The following chapter demonstrates this by the example of an inkjet printing device.

Chapter 5

Application of the code - Inkjet printer

Besides the more academical examples, which we used to demonstrate the abilities and limitations of the code, in this chapter a first principal application of the code to an close-to-industry problem is presented: The drop ejection for inkjet printing

In inkjet printing processes, the liquid ink is pushed out of the nozzle by a piezo-electric device. For the best printing performance, it is necessary to get very small droplets ejected from the nozzle, whose size is as regular as possible. Later in the printing rocess, the drops will be deflected by an electro-magnetic fields, in which the drop size plays an important role on the final deflection. This is why inkjet manufactures are interested into simulating the typical formation of satellite drops, which result from the separation of the main drop from the reservoir.

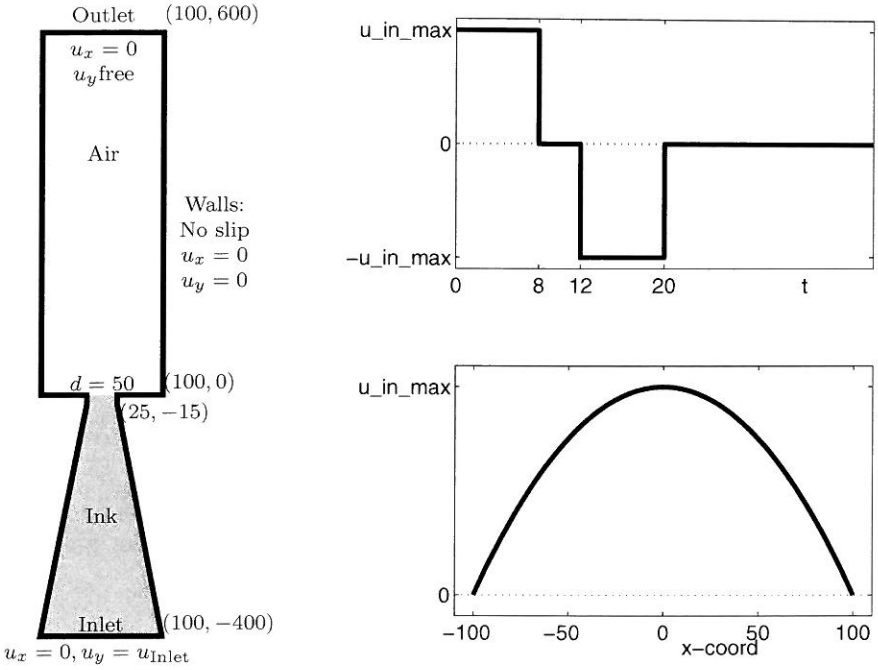
In the publications of (Yu et al., 2005) and (Wu et al., 2004), detailed investigations are published on this topic. These papers also use a level set approach on a two phase flow of ink and air. Furthermore, the simulations published are also used to show the effect of the nozzle shape for the ink drops. A crucial point on this seems to

be the model of the boundary conditions between the ink and the boundary close to the interface. More advanced models, including slip- and separation-angles are demonstrated. For the code developed here, this is beyond the scope. For first, it is only applied to the inkjet setting to prove the general ability of the code to cope those problems.

We use a simplified geometry of the nozzle and are only interested in the separation of drops from the ink reservoir. In Figure (5.1(a)) the basic setting we used is shown. The general shape is adapted from (Wu et al., 2004) and then transferred into the dimensions used in (Yu et al., 2005).

Although it could be seen as a axis-symmetric problem, we chose to model it in two dimensions, because of the inaccuracies of the symmetry condition of the surface tension we experienced in the rising bubbles problems. The full geometry is symmetric, and the actual nozzle diameter is about $d = 50$. The water-ink-interface is initialized at $y = 0$. As boundary conditions, we imposed zero velocity on the walls. On the outlet, we imposed a vertical velocity ($u_x = 0$) and did not prescribe the y-velocity u_y.

As inflow conditions, we simplified the physics to a velocity profile for the maximum inflow velocity at the center bottom. As inflow y-velocity u_y we choose a time-dependent velocity, which first pushes fluid into the domain, then a short period with zero inflow velocity and finally a period of negative inflow velocity, which pulls the ink back into the reservoir and forces the ink drop separation. In Figure (5.1(b)) the $u_y(t)$-profile is shown for the center of the nozzle bottom. From this center, a parabolic shape is chosen on the bottom to fit the no-slip boundary condition on the side walls of the domain. The publications mentioned above include an oscillation model for the piezo-electric actuation of the ink on the bottom or a pressure profile as inflow condition. Since this is not available in the current code, we had to find a maximum inflow velocity. The results we show here, are gained with a maximum inflow velocity of $u_{\text{Max Inflow}} = 2.6\frac{\mu m}{\mu s}$.



(a) Geometry

(b) Inflow velocity over time and bottom x-coordinate

Figure 5.1: Geometry and inflow conditions of the inkjet problem

Viscosity:	$\mu_1 =$	$3.34 \cdot 10^{-3} \frac{kg}{m\,s}$	$=$	$3.34 \cdot 10^{-6} \frac{\mu g}{\mu m\,\mu s}$
	$\mu_2 =$	$1.775 \cdot 10^{-3} \frac{kg}{m\,s}$	$=$	$1.776 \cdot 10^{-6} \frac{\mu g}{\mu m\,\mu s}$
Density:	$\rho_1 =$	$1070 \frac{kg}{m^3}$	$=$	$1.07 \cdot 10^{-6} \frac{\mu g}{(\mu m)^3}$
	$\rho_2 =$	$1.225 \frac{kg}{m^3}$	$=$	$1.225 \cdot 10^{-9} \frac{\mu g}{(\mu m)^3}$
Surface tension	$\tau =$	$0.032 \frac{kg}{s^2}$	$=$	$32 \cdot 10^{-6} \frac{\mu g}{(\mu s)^2}$

Table 5.1: Physical parameters of ink (ρ_1, μ_1) and air (ρ_2, μ_2) in the correct dimensions

The problem is meant to happen on a microscopic scale, so the physical parameters, which we took from (Wu et al., 2004) have to be converted into the corresponding dimensions. We choose micrometer (μm), microseconds (μs) and $1 \cdot 10^{-6}$ microgramms ($1 \cdot 10^{-6} \mu g$) in order to have proper order of numbers. Table 5.1 shows the conversion of the used values.

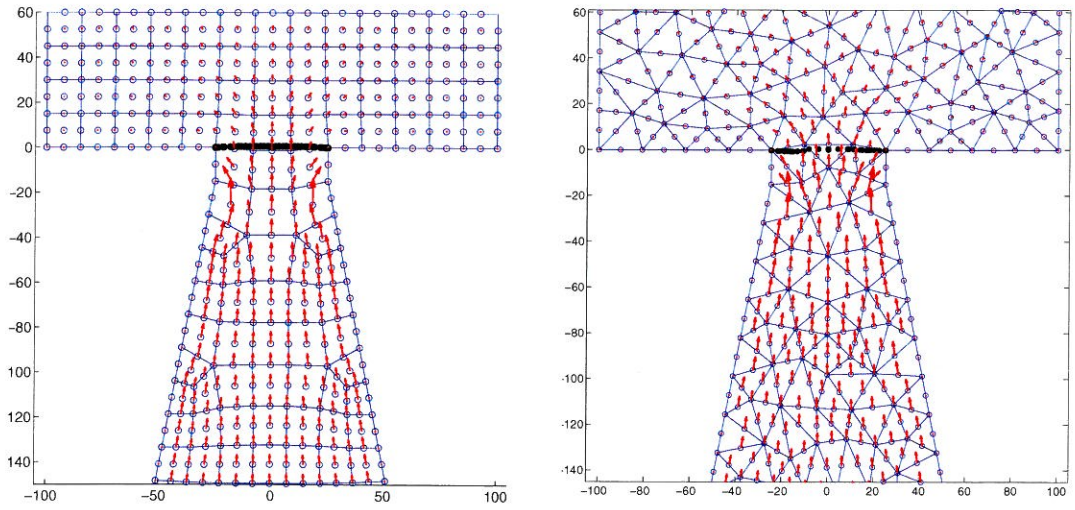


Figure 5.2: Quadrilateral and triangular mesh and interface of the inkjet case after the first time step, including the velocity

The mesh for the first calculation of the inkjet problem is a relatively coarse Q2Q1 mesh, produced with EZ4U, an academic free mesh generation software produced by the LaCàN (LaCàN, 2010). Secondly, a triangular P2P1 mesh with approximately the same number of degrees of freedom is used for this example. In Figure (5.2) a close-up of both meshes close to the ejection is shown. As one can see, the meshes are regular and do not contain refinements. Further investigations on this example should include some studies on the effect of the mesh refinement in the ejection area.

5.1 Results of the inkjet application on a quadrilateral mesh

For the correct values, we gained an inkjet drop, which finally could not completely separate from the ink reservoir. Figure (5.3) shows for several stages, how the interface evolves on this. The inflow expands the interface out of the nozzle. A drop forms during the period with zero inflow velocity. The pulling negative inflow velocity tries

to separate the drop from the reservoir and pulls the two fluids back. With this, a long slender neck connects the drop with the reservoir, and the drops inertia is strong enough to move it forward away from the reservoir. But in contrast to the physical expectations, the neck does not get thinner while the drop is moving away from the reservoir. Therefore the drop can not separate from the reservoir, but stays connect.

The tendency that those thin layers of one fluids are kept longer than it would be physical, has already been observed for the joining bubbles example, where the separating layer between the bubbles and the free surface stayed longer than in the computations of others. For now, it is hard to determine the reason for this and further investigations are required. Especially finer meshes, which are better able to resolve the fine geometrical aspects of the level set and therefore of the interface, should improve those problems.

5.2 Results of the inkjet application on a triangular mesh

For the same settings of the inkjet drop ejection process, we used a triangular mesh. The number of degrees of freedom is approximately the same and this mesh is also not refined. In contrast to the calculation on the quadrilateral mesh, we had been able to get a drop separation from the reservoir, as the sequence in Figure (5.4) shows. For this mesh, the code was also able to reproduce one satellite drop, which is formed out of the neck. By this, it may be assumed that with high mesh refinement, a more detailed and accurate droplet ejection process could be modeled.

This second example shows, how much at least for coarser mesh the solution depends on the meshes. Especially by the interface representation with the points of the reference mesh, a strong mesh influence is introduced. Further studies on this are required. But this first two attempts show that the code is in general able to resolve the effects of droplet ejection in inkjet printing processes qualitatively.

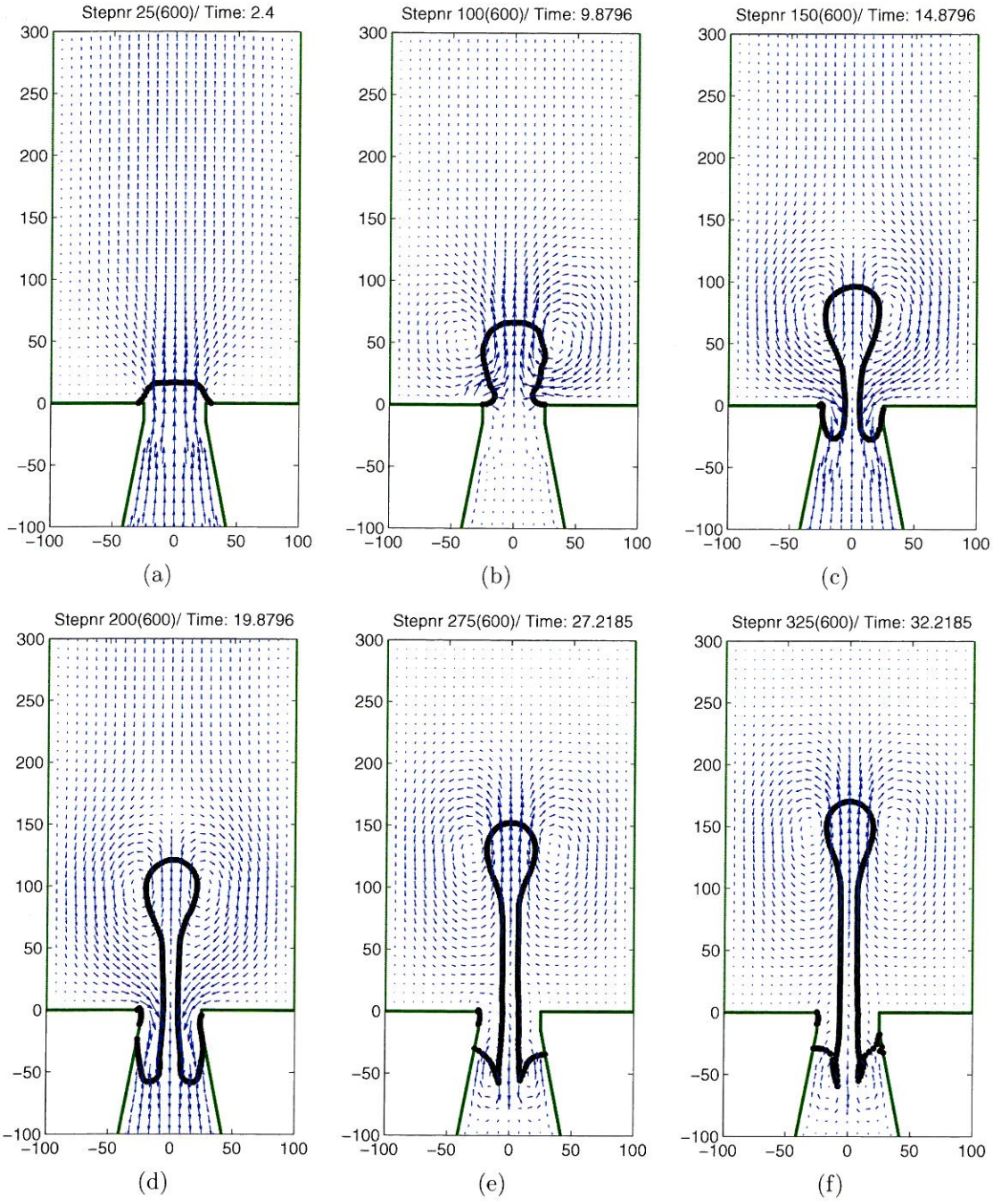


Figure 5.3: Velocity and interface shape for inkjet application with $u_{\text{Max Inflow}} = 2.6\frac{\mu m}{\mu s}$ on a quadrilateral mesh

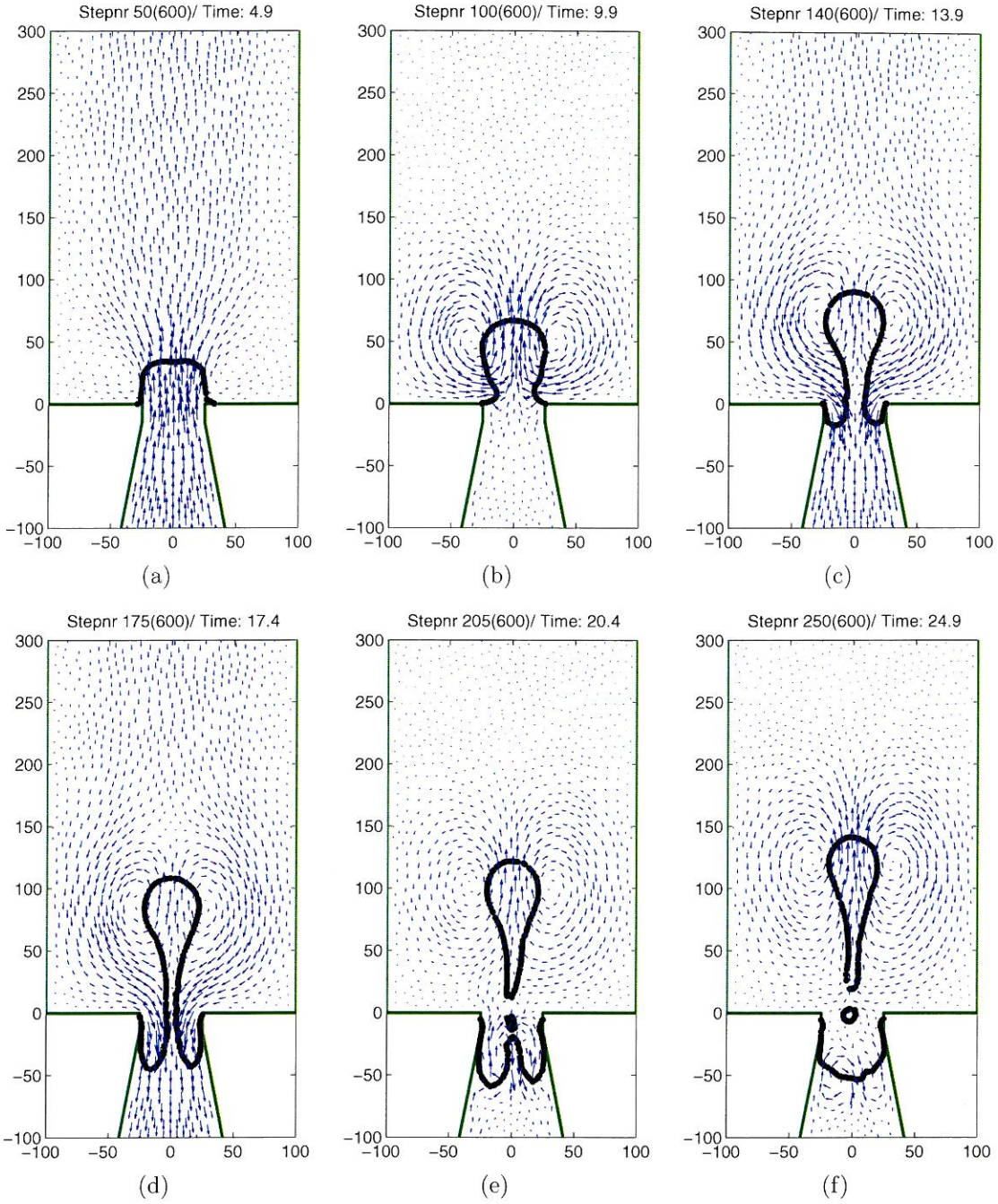


Figure 5.4: Velocity and interface shape for inkjet application with $u_{\text{Max Inflow}} = 2.6\frac{\mu m}{\mu s}$ on a triangular mesh

Chapter 6

Summary and Outlook

6.1 Summary of the code and results

In the previous chapters, we presented a method two simulate immiscible two phase flows with surface tension. The code solves the incompressible Navier-Stokes equations, including a localized forcing term for the surface tension. The surface tension itself is dependent on the physical surface tension coefficient and the curvature of the interface.

The equations are solved in the weak form, with an extended finite elements method, which is due to discontinuous enrichment able to fully represent the non-smooth discontinuity of the velocity and the jump discontinuity of the pressure. As time integration scheme, a semi-implicit method is chosen, which needs stabilization of the convection term. The integration of the split elements is realized with reference triangulation. The curvature of the interface is determined by using a piecewise polynomial approximation of the interface position. We also presented how the continuous front tracking method with a level set function is used to localized the interface position, including the aspects of the maintenance like smoothing and re-initialization.

The method was applied to many examples, each of it to demonstrate different aspects of the code. A steady non-physical example shows the improvements in the discontinuous solution of velocity and pressure by the enrichment. Oscillations close to the interface nearly vanish due to the enrichment, and the mesh convergence rate is increased from first to second order.

A reversible rotating Stokes flow presents the different aspects of moving the level set. It gives a first hint that mass conservation is first order accurate by mesh refinement and time step size. Also, in this example we showed that the level set maintenance corrupts the mass conservation properties of the scheme at that they should be used as less as possible. This example also was used to demonstrate the effect of surface tension in general.

With a Rayleigh-Taylor-instability example, we showed the first comparison of our code to others. We can state a qualitatively good behavior of our code for this example. But it was also shown that due to the polynomial approximation of the interface, we introduce non-symmetric conditions, even for a symmetric set-up.

Further comparisons on the accuracy of the code have been made with rising bubble examples. Here again, the regularizing effect of the surface tension was proved and compared with results from numerical and physical experiments, a general agreement was shown. But for examples with higher Reynolds numbers, the method presented here showed its limitations: While moving the level set, non-physical oscillations appear and corrupt the solution.

A main intention while developing the code was to be able to handle high changes in the topology of the interface. This ability was successfully demonstrated with an example of two bubbles rising to a free surface. But also, it appears that thin aspects of the interface, seem to stay longer in the solution than in other codes.

Finally, as an outlook on the possible technical applications of the code, a drop ejection process of inkjet printers is simulated. Here we were able to demonstrate

the codes' abilities of handling drop separation, including the formation of a satellite drop.

In all the examples, but especially for the rising bubbles and the inkjet, stability sets narrow limitations of the examples the code can be applied to. The Reynolds number and the Bond number have to be relatively small, otherwise oscillations of the level set could corrupt the solution. But despite this limitations in the stability, we had been able to present a method that is able to simulate two-phase flows properly. The important effects of the surface tension are captured correctly, including the problematic discontinuities of velocity and pressure. For a huge variety of examples, qualitatively good results with proper mass conservation properties have been shown.

6.2 Future aspects

6.2.1 Investigations on the accuracy

As mentioned during the validation in Chapter 4, the results shown here are just the first calculations on the problem, with relatively coarse meshes and without further investigation of the parameters that could influence the results. Especially for the more complicated cases of the joining bubbles and the inkjet application, more calculation, with more specific meshes are necessary to specify the results.

6.2.2 Level Set: Time Stepping, Curvature, Re-Initialization

As already mentioned, the stability of the code would be the next thing to deal with. The main problem seems to be the oscillations of the level set, even if time step size is far below the stability condition of the TG3-2S time stepping scheme. Other high-order accurate, but possibly implicit methods may be tested, to show whether they

are better able to handle those problems. For this, it is always difficult to find the good trade-off between stability and accuracy, which is in our case mainly the mass conservation.

Secondly, on the examples on the rising bubbles and Rayleigh-Taylor instability we observed some inaccuracies on the surface tension, especially for the axis symmetric cases. Our attempt to approximate the interface position with a piecewise polynomial is rather unique. Other papers determine the curvature by the divergence of the interface normal direction. Also, there exists attempts to approximate the interface position and curvature with splines of higher order, in which the curvature is continuous. Tests on those methods would help to determine the advantages and disadvantages of each of the techniques to determine the level set position and curvature.

Also, the way the reinitialization of the level set was implemented here, is relatively coarse, with the bad side effect of changing the position of the interface. This does not fulfill the requirement of mass conservation. More sophisticated methods would be able to solve this problem better.

Shortly before the end of the preparation of this thesis, a "Toolbox of Level Set Methods" draw attention (Mitchell, 2010, 2008), which includes several methods of time stepping, curvature-dependent movement and mass-conserving reinitialization. It could be worth a try to include aspects of this toolbox into our code to see the results.

6.2.3 Mass conserving correction of interface position

Since mass conservation is one of the most important issues on the two-phase flows, there exist methods that enforce the mass conservation by a correction step. For example, in (Hua and Lou, 2007), such a correction is applied in the background of rising bubbles examples.

6.2.4 Computing time: parallelization

To make the computations more efficient, parallelization of several aspects of the calculation would be possible. Currently, for the integration, loops over all elements are implemented, which do not influence each other and easily could be performed in parallel. Solving the system for velocity and pressure is done with the Matlab-implemented matrix solvers, for which parallel options normally are available. For the examples of this thesis, meshes are coarse enough, that it would have been possible, to let the calculations run on a standard Microsoft Windows laptop with 3GB of memory in a reasonable time. Anyways, for finer meshes and/or much more time steps, it would be very advantageous to have the code prepared for parallel computing.

Bibliography

Bhaga, D. and M. E. Weber (1981). Bubbles in viscous liquids: shapes, wakes and velocities. *Journal of Fluid Mechanics 105*, 61–85.

Chessa, J. and T. Belytschko (2003). An enriched finite element method and level sets for axisymmetric two-phase flow with surface tension. *International Journal for Numerical Methods in Engineering 58*, 2041–2064.

de Sousa, F., N. Mangiavacchi, L. Nonato, A. Castelo, M. Tomé, V. Ferreira, J. Cuminato, and S. McKee (2004). A front-tracking/front-capturing method for the simulation of 3d multi-fluid flows with free surfaces. *Journal of Computational Physics 198*(2), 469–499.

Donea, J. and A. Huerta (2003). *Finite Element Methods for Flow Problems*. Chichester, West Sussex PO19 8SQ, England: Wiley.

Groß, S. (2008). *Numerical methods for three-dimensional incompressible two-phase flow problems*. Ph. D. thesis, RWTH Aachen University.

Groß, S., V. Reichelt, and A. Reusken (2006). A finite element based level set method for two-phase incompressible flows. *Comput. Vis. Sci. 9*(4), 239–257.

Hermann, S. and R. Klette (2007). A comparative study on 2d curvature estimators. In *ICCTA '07: Proceedings of the International Conference on Computing: Theory and Applications*, Washington, DC, USA, pp. 584–589. IEEE Computer Society.

Hua, J. and J. Lou (2007). Numerical simulation of bubble rising in viscous liquid. *Journal of Computational Physics 222*(2), 769 – 795.

LaCàN (2010). EZ4U. mesh generation environment. `http://www-lacan.upc.edu/ez4u.htm` (accessed May 31st, 2010). Laboratori de Càlcul Numèric, Universitat Politècnica de Catalunya.

Li, Z., K. Ito, and M.-C. Lai (2007). An augmented approach for stokes equations with a discontinuous viscosity and singular forces. *Computers & Fluids 36*(3), 622 – 635.

Lock, N., M. Jaeger, M. Medale, and R. Occelli (1998). Local mesh adaptation technique for front tracking problems. *Int. J. Numer. Meth. Fluids 28*, 719–736.

Marchandise, E., P. Geuzaine, N. Chevaugeon, and J.-F. Remacle (2007). A stabilized finite element method using a discontinuous level set approach for the computation of bubble dynamics. *Journal of Computational Physics 225*(1), 949 – 974.

Mitchell, I. (2010). A toolbox of level set methods. `http://www.cs.ubc.ca/~mitchell/ToolboxLS/`, accessed May, 23rd, 2010. Department of Computer Science, University of British Columbia.

Mitchell, I. M. (2008). The flexible, extensible and efficient toolbox of level set methods. *Journal of Scientific Computing 45*(2–3), 300–329.

Moes, N., J. Dolbow, and T. Belytschko (1999). A finite element method for crack growth without remeshing. *International Journal for Numerical Methods in Engineering 46*(1), 131–150.

Moës, N., M. Cloirec, P. Cartraud, and J. F. Remacle (2003). A computational approach to handle complex microstructure geometries. *Computer Methods in Applied Mechanics and Engineering 192*(28-30), 3163 – 3177. Multiscale Computational Mechanics for Materials and Structures.

Puckett, E. G., A. S. Almgren, J. B. Bell, D. L. Marcus, and W. J. Rider (1997). A high-order projection method for tracking fluid interfaces in variable density incompressible flows. *Journal of Computational Physics 130*(2), 269 – 282.

Smolianski, A. (2001). *Numerical Modeling of Two-Fluid Interfacial Flows.* Ph. D. thesis, University of Jyväskylä. ISBN 951-39-0929-8.

Tornberg, A.-K. and B. Enquist (2000). A finite element based level-set method for multiphase flow applications. *Computing and Visualization in Science 3*, 93 – 101.

Wu, H.-C., H.-J. Lin, Y.-C. Kuo, and W.-S. Hwang (2004). Simulation of droplet ejection for a piezoelectric inkjet printing device. *Materials transactions - Japan Institute of Metals 45*(3), 893–899.

Yu, J.-D., S. Sakai, and J. Sethian (2005). A coupled quadrilateral grid level set projection method applied to ink jet simulation. *Journal of Computational Physics 206*(1), 227 – 251.

Zlotnik, S. (2008). *Numerical modeling of transient multiphase thermo-mechanical problems: Application to the oceanic lithosphere.* Ph. D. thesis, Departament de Geodinàmica i Geofísica, Facultat de Geologia, Universitat de Barcelona.